

SSSSSSSS SSSSSSSS AAAAAAA AAAAAAA TTTTTTTTTT TTTTTTTTTT SSSSSSSS SSSSSSSS SSSSSSSS SSSSSSSS 000000 000000 5555555555
SS AA AA TT SS SS SS SS 00 00 55
SS AA AA TT SS SS SS SS 00 00 55
SS AA AA TT SS SS SS SS 00 00 55
SSSSSS SSSSSS AA AA TT SSSSSS SSSSSS SSSSSS SSSSSS 00 00 55
SSSSSS SSSSSS AA AA TT SSSSSS SSSSSS SSSSSS SSSSSS 00 00 55
SS AA AA TT SS SS SS SS 00 00 55
SS AA AA TT SS SS SS SS 00 00 55
SS AA AA TT SS SS SS SS 00 00 55
SSSSSSSS SSSSSSSS AA AA TT SSSSSSSS SSSSSSSS SSSSSSSS SSSSSSSS 000000 000000 55555555
SSSSSSSS SSSSSSSS AA AA TT SSSSSSSS SSSSSSSS SSSSSSSS SSSSSSSS 000000 000000 55555555
LL IIIIIII SSSSSSSS
LL IIIIIII SSSSSSSS
LL IIIIIII SS SS
LLLLLLLLLL IIIIIIII SSSSSSSS
LLLLLLLLLL IIIIIIII SSSSSSSS

(1)	57	DECLARATIONS
(1)	236	R/W PSECT
(1)	390	SATSSS05
(1)	439	SNDACC TESTS
(1)	514	SNDERR_S TESTS
(2)	565	SNDOPR_TESTS
(2)	720	SND SMB TESTS
(2)	980	REG_SAVE
(2)	1001	REG_CHECK
(2)	1043	PRINT_FAIL
(2)	1089	READ_CHECK
(2)	1129	CRE_JOB
(2)	1215	BUF_CHECK
(2)	1266	SND_CHECK
(2)	1306	GENREQ
(2)	1329	MODE_ID

0000 1 .TITLE SATSSS05 - SATS SYSTEM SERVICE TESTS (SUCC S.C.)
0000 2 .IDENT 'V04-000'
0000 3 .
0000 4 .
0000 5 *****
0000 6 *
0000 7 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9 * ALL RIGHTS RESERVED.
0000 10 *
0000 11 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16 * TRANSFERRED.
0000 17 *
0000 18 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20 * CORPORATION.
0000 21 *
0000 22 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24 *
0000 25 *
0000 26 *****
0000 27 .
0000 28 .
0000 29 ++
0000 30 :FACILITY: SATS SYSTEM SERVICE TESTS
0000 31 :ABSTRACT: The SATSSS05 module tests the execution of the following
0000 32 : VMS system services:
0000 33 :
0000 34 :
0000 35 :\$SNDACC
0000 36 :\$SNDERR
0000 37 :\$SNDOPR
0000 38 :\$SNDSMB
0000 39 :
0000 40 :
0000 41 :ENVIRONMENT: User mode image.
0000 42 : Needs CMKRNL privilege and dynamically acquires other
0000 43 : privileges, as needed.
0000 44 :
0000 45 :AUTHOR: Larry D. Jones. CREATION DATE: JULY, 1978
0000 46 :
0000 47 :MODIFIED BY:
0000 48 :
0000 49 : V03-002 PCG0001 Peter C. George 16-Feb-1981
0000 50 : Add OPCMSG macro expansion
0000 51 :
0000 52 : V03-001 LDJ0001 Larry D. Jones 17-Sep-1980
0000 53 : Modified to conform to new build command procedures.
0000 54 :**
0000 55 :--

```
0000 57 .SBTTL DECLARATIONS
0000 58 : MACRO LIBRARY CALLS
0000 60 :
0000 61 $ACCDEF : accounting definitions
0000 62 $DIBDEF : device info block offsets
0000 63 $EMBDEF : error log buffer definitions
0000 64 $JBCMSGDEF : job controller definitions
0000 65 $OPCDEF : operator communications def.
0000 66 $OPCMMSG : operator communications messages
0000 67 $OPRDEF : operator message definitions
0000 68 $PHDDEF : process header definitions
0000 69 $PRVDEF : privilege definitions
0000 70 $SHR MESSAGES UETP,116,<<TEXT,INFO>> : UETP$ TEXT definition
0000 71 $SMRDEF : symbiot manager definitions
0000 72 $STSDEF : STS definitions
0000 73 $UETPDEF : UETP message definitions
0000 74 :
0000 75 : Equated symbols
0000 76 :
00000000 0000 77 WARNING = 0 : warning severity value for msgs
00000001 0000 78 SUCCESS = 1 : success "
00000002 0000 79 ERROR = 2 : error "
00000003 0000 80 INFO = 3 : information "
00000004 0000 81 SEVERE = 4 : fatal "
00000000D 0000 82 :
0000000A 0000 83 CR = 13 : terminal definitions
0000000A 0000 84 LF = 10 :
00000006 0000 85 :
00000006 0000 86 FIDSIZ = 6 : ID sizes
00000006 0000 87 DIDSIZE = 6 :
00000014 0000 88 FILNAMSIZ = 20,
00000007 0000 89 COM_FIL_SIZ = 7,
00000064 0000 90 :
00000064 0000 91 BUF_SIZE=100 : buffer size
00000064 0000 92 :
00000064 0000 93 ALL_OPR = OPCSM_NM_CENTRL!OPCSM NM PRINT!-
00000064 0000 94 OPCSM_NM_TAPES!OPCSM NM DISKS!-
00000064 0000 95 OPCSM_NM_DEVICE!OPCSM NM OPER1!-
00000064 0000 96 OPCSM_NM_OPER2!OPCSM NM OPER3!-
00000064 0000 97 OPCSM_NM_OPER4!OPCSM NM OPER5!-
00000064 0000 98 OPCSM_NM_OPER6!OPCSM NM OPER7!-
00000064 0000 99 OPCSM_NM_OPER8!OPCSM NM OPER9!-
00000064 0000 100 OPCSM_NM_OPER10!OPCSM NM OPER11!-
00000064 0000 101 OPCSM_NM_OPER12
0000001F 0000 102 :
0000001F 0000 103 : ***** NOTE *****
00000008 0000 104 :
00000008 0000 105 : THE FOLLOWING DEFINITION IS TO BE REMOVED WHEN VMS RELEASE 2 IS FIXED.
00000008 0000 106 :
00000008 0000 107 SNDACCS_CHAN = 8
00000008 0000 108 : MACROS
00000008 0000 109 :
```

SATSSS05
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) L⁷ 16-SEP-1984 00:46:10 VAX/VMS Macro V04-00
DECLARATIONS 5-SEP-1984 04:29:47 [UETPSY.SRC]SATSSS05.MAR;1 Page 3 (1)

00000000 111 .PSECT RODATA,RD,NOWRT,NOEXE,LONG
0000 112 .TEST_MOD_NAME:
0000 113 :ASCIIC /SATSSS05/ : needed for SATSMS message
0000 114 :ASCIID /SATSSS05/
0009 115 TEST_MOD_NAME_D:
0009 116 :ASCIID /SATSSS05/ : module name
0017 117 TEST_MOD_BEGIN:
0019 118 :ASCIIC /begun/
001F 119 TEST_MOD_SUCC:
001F 120 :ASCIIC /successful/
002A 121 TEST_MOD_FAIL:
002A 122 :ASCIIC /failed/
0031 123 SNDACC:
0031 124 :ASCIIC /SNDACC/
0038 125 SNDERR:
0038 126 :ASCIIC /SNDERR/
003F 127 SNDOPR:
003F 128 :ASCIIC /SNDOPR/
0046 129 SND SMB:
0046 130 :ASCIIC /SND SMB/
004D 131 CS1:
004D 132 :ASCIID \Test !AC service name !AC step !UL failed.\
007F 133 CS2:
007F 134 :ASCIID \Expected !AS = !XL received !AS = !XL\
00AC 135 CS3:
00AC 136 :ASCIID \Expected !AS!UB = !XL received !AS!UB = !XL\
00DF 137 CS5:
00DF 138 :ASCIID \Mode was !AS.\
00F4 139 CS6:
00F4 140 :ASCIID \Expected byte offset !UB(10) = !XB(16) received !XB(16).\
0134 141 UM:
0134 142 :ASCIID \user\

```

42 4D 24 54 53 53 00000148'010E0000' 0140 143 MBNAM: .ASCID \$SSTSMBX\
      5B 0140 144 .ASCID \$SSTSMBX\
      014E 145 TTNAME: .ASCLC \_TTA\
      014F 04 014F 146 .ASCLC \_TTA\ ; terminal name to send opr messages to
      0154 0001 0154 147 TTUNIT: .WORD 1 ; unit number for above
      0156 0156 148 EXP: .ASCID \status\
      0164 0164 150 .ASCID \status\
      0172 0172 151 BAT_IMP_EXC: .ASCLC \Batch job improperly executed.\ ; batch job symbol name
      017E 018A 152 .ASCLC \Batch job improperly executed.\ ; parameter for SNDMSG
      00000003 018A 153 YES_DESC: .LONG 3
      00000192' 018E 154 .ADDRESS SYM_NAME
      0192 0192 155 .ADDRESS SYM_NAME ; batch job symbol name
      4D 59 53 0192 156 SYM_NAME: .ASCII \SYM\
      0195 0195 157 .ASCII \SYM\
      00000014 0195 158 SYM_DESC: .LONG 20
      0000038A' 0199 159 .ADDRESS SYM
      019D 019D 160 .ADDRESS SYM ; SNDMSG test data
      53 45 59 00' 019D 161 YES: .ASCLC \YES\
      03 019D 162 .ASCLC \YES\ ; parameter for SNDMSG
      01A1 163 QUENAM1: .ASCLC /UETP_BAT_QUE1/
      01A1 164 .ASCLC /UETP_BAT_QUE1/
      55 51 5F 54 41 42 5F 50 54 45 55 00' 01A1 165 QUENAM1L=-QUENAM1 ; PUTMSG message vector
      31 45 0D 01AD 01A1 166 QUENAM2: .ASCLC /UETP_BAT_QUE2/
      0000000E 01AF 167 .ASCLC /UETP_BAT_QUE2/
      01BD 01BD 168 QUENAM2L=-QUENAM2 ; SNDERR test data
      00000003 01BD 169 MSGVEC: .LONG 3
      00741133 01C1 170 .LONG UETPS_TEXT
      00000001 01C5 171 .LONG 1
      00000169' 01C9 172 .ADDRESS MESSAGE1
      01CD 01CD 173 .ADDRESS MESSAGE1 ; GENREQ routine OPRMSG buffer
      00000064 01CD 174 TEST_ERROR: .LONG BUF_SIZE
      000001D5' 01D1 175 .ADDRESS :+4
      00000000 01D5 176 A=0
      01D5 01D5 177 .REPT BUF_SIZE
      01D5 01D5 178 .BYTE A
      01D5 01D5 179 A=A+1
      00 01D5 180 .ENDR
      0239 0239 181 OPNAME: .ASCLC /_OPA/
      41 50 4F 5F 00' 0239 182 OPNAME: .ASCLC /_OPA/ ; operator console name
      04 0239 183 .ASCLC /_OPA/
      023E 023E 184 OP_MSG1: .LONG MSG1L ; request operator type
      00000036' 023E 185 .ADDRESS :+4 ; is only 3 bytes big
      00000246' 0242 186 .BYTE OPCS_RQ_RQST
      03 0246 187 .LONG OPCSM_NM_CENTRL
      00000001 0247 188 .=-1
      0000024A 024B 189
  
```

SATSSS05
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:46:10 VAX/VMS Macro V04-00
DECLARATIONS N 7 5-SEP-1984 04:29:47 [UETPSY.SRC]SATSSS05.MAR;1

Page 5
(1)

00000000 024A 190 .LONG 0 : global request ID of 0
52 50 4F 44 4E 53 24 20 50 54 45 55 024E 191 OP_MESG:
76 72 65 73 20 6D 65 74 73 79 73 20 024E 192 .ASCII /UETP \$SNDOPR system service test user message./
65 73 75 20 74 73 65 74 20 65 63 69 025A 193 OP_MESG_LEN=.-OP_MESG
2E 65 67 61 73 73 65 6D 20 72 0266 194 MSG1L=.=OP_MSG1-8 : message buffer size
0000002E 027C 195 FILE_NAME:
00000036 027C 196 .ASCII /S05.COM/
4D 4F 43 2E 35 30 53 00' 027C 197 NAME_SIZE=.-FILE_NAME
07 027C 198 .BLKB <FILNAM\$IZ-NAME_SIZE> ; filler for SND\$MB
00000008 0284 199 FILE_NAME1:
00000290 0284 200 .ASCII /S05.LOG/ ; log file name
47 4F 4C 2E 35 30 53 0290 201 COM_FILE:
20 35 30 53 53 53 54 41 53 20 21 24 0297 202 .ASCII /\$! SATSSS05 SND\$MB test batch job/<CR><LF>
20 74 73 65 74 20 42 4D 53 44 4E 53 02A3
0A 0D 62 6F 6A 20 68 63 74 61 62 02AF
21 24 02BA 203 .ASCII /\$/!
00000025 02BC 204 RECO_SIZE=.-COM_FILE : record 0 size
27 31 50 27 3D 3A 4D 59 53 20 24 02BC 205 REC1:
00000008 02C7 206 .ASCII /\$ SYM:='P1'/
53 51 45 2E 4D 59 53 20 46 49 20 24 02C7 207 REC1_SIZE=.-REC1 : record 1 size
20 4E 45 48 54 20 22 53 45 59 22 2E 02D3
27 20 4D 59 53 20 52 47 2F 46 45 44 02DF
0A 0D 27 31 50 02EB 208 REC2:
00000029 02F0 209 .ASCII \\$ IF SYM.EQS."YES" THEN DEF/GR SYM 'P1'\<CR><LF>
02F0 210 REC2_SIZE=.-REC2
07' 02F0 211 OL1:
21 02F1 212 .BYTE OL1\$
26 02F2 213 .BYTE SMO\$K_HOLD
53 45 59 00' 02F3 214 .BYTE SMO\$K_PARAMS
03 02F3 215 .ASCII /YES/
00 02F7 216 .BYTE 0
00000007 02F8 217 OL1\$=.-OL1-1
02F8 218 OL2:
04' 02F8 219 .BYTE OL2\$
22 02F9 220 .BYTE SMO\$K_JOBPRI
03 02FA 221 .BYTE 3
21 02FB 222 .BYTE SMO\$K_HOLD
00 02FC 223 .BYTE 0
00000004 02FD 224 OL2\$=.-OL2-1
31 4D 55 4E 5F 42 4F 4A 00' 02FD 225 JN1:
08 02FD 226 .ASCII /JOB_NUM1/
00 0306 227 JN2:
32 4D 55 4E 5F 42 4F 4A 00' 0307 228 .BYTE 0
08 0307 229 .ASCII /JOB_NUM2/
00 0310 230 JN3:
33 4D 55 4E 5F 42 4F 4A 00' 0311 231 .BYTE 0
08 0311 232 .ASCII /JOB_NUM3/
00 031A 233 .BYTE 0

```

031B 235 :          .SBTTL R/W PSECT
031B 236 :PSECT RWDATA,RD,WRT,NOEXE,LONG
00000000 237 :PSECT RWDATA,RD,WRT,NOEXE,LONG
00000000 238 :PID:
00000000 239 :PID: LONG 0 ; PID for this process
00000000 240 :CURRENT_TC: LONG 0 ; ptr to current test case
00000000 241 :.LONG 0
00000000 242 :.ALIGN LONG
00000000 243 :REG_SAVE_AREA: BLKL 15 ; register save area
00000044 244 :MOD_MSG_CODE: LONG UETPS_SATSMS ; test module message code for putmsg
007480D9 245 :.BLKL 15
0044 246 :.LONG UETPS_SATSMS
00000000 247 :TMN_ADDR: ADDRESS TEST_MOD_NAME
00000000 248 :.ADDRESS TEST_MOD_NAME
00000000 249 :TMD_ADDR: ADDRESS TEST_MOD_BEGIN
00000019 250 :.ADDRESS TEST_MOD_BEGIN
00000000 251 :PRVPRT: BYTE 0 ; protection return byte for SETPRT
00000000 252 :PRIVMASK: QUAD 0 ; priv. mask
00000000 253 :CHM_CONT: LONG 0 ; change mode continue address
00000000 254 :RETADR: BLKL 2 ; returned address's from SETPRT
00000065 255 :.QUAD 0
00000065 256 :STATUS: LONG 0
00000000 257 :.LONG 0
00000065 258 :.BLKL 2
00000000 259 :.LONG 0
00000000 260 :QIO: $QIO 2,MBCHAN,IOS_READVBLK,,,BUF,BUF_SIZE+30 ; QIO parameter list
00000000 261 :$SNDACC ACC_DESC,MBCHAN ; SNDACC parameter list
000A9 262 :$SNDE: $SNDERR TEST_ERROR ; SNDERR parameter list
000A9 263 :$SNDOPR OPMMSG_DESC,0 ; SNDOPR parameter list
000B1 264 :$SNDOPR OPMSG_DESC,0 ; SNDOPR parameter list
000B1 265 :$SNDNSMB SMSG_DESC,0 ; SNDNSMB parameter list
000BD 266 :$SNDNSMB SMSG_DESC,0 ; SNDNSMB parameter list
000BD 267 :REG: ASCID \register R\ ; register number
00000000 268 :.LONG 0
00000000 269 :.LONG 130 ; buffer desc.
00000082 270 :.ADDRESS BUF
000000E7 271 :.BLKB 130
000000E7 272 :MESSAGEL: LONG 0 ; message desc.
00000000 273 :.ADDRESS BUF
00000000 274 :SERV_NAME: LONG 0 ; service name pointer
00000000 275 :.LONG 0
00000000 276 :MBCHAN: WORD 0 ; mailbox channel number
00000000 277 :.WORD 0
00000000 278 :MODE: LONG 0 ; current mode string pointer
00000000 279 :.LONG 0
00000000 280 :.WORD 0
00000000 281 :.LONG 0
00000000 282 :.ADDRESS BUF
00000000 283 :.BLKB 130
00000000 284 :MESSAGEL: LONG 0 ; message desc.
00000000 285 :.ADDRESS BUF
00000000 286 :.BLKB 130
00000000 287 :SERV_NAME: WORD 0 ; service name pointer
00000000 288 :.WORD 0
00000000 289 :MODE: LONG 0 ; current mode string pointer
00000000 290 :.WORD 0

```

```

000001CB 017B 291 .BLKB 80 ; mailbox buffer
00000003 01CB 292 MSGVEC1: ; PUTMSG message vector
00741133 01CF 293 .LONG 3
00000001 01D3 294 .LONG UETPS_TEXT
00000000 01D7 295 .LONG 1
00000000 01DB 296 .LONG 0
000001E3 01DB 297 STATUS: ; mailbox status block
000001E3 01E3 298 .BLKL 2
0001 01E3 299 ACC_MSG: ; starting message code
0052 01E5 300 .WORD ACC$K_INSMESG
00000001 01E7 301 .WORD MSG_SIZE ; message size
00000000 01EB 302 .LONG 1 ; final exit status
00000002 01EF 303 .LONG 0 ; PID
00000000 01F3 304 .LONG 2 ; job ID
54 53 45 54 53 59 53 00 01FB 305 .QUAD 0 ; system job termination time
07 01FB 306 .ASCIC /SYSTEST/ ; account name
43 41 44 4E 53 24 20 50 54 45 55 00 0203 307 .ASCIC /UETP $SNDACC system service test user data record/ ; user data
72 65 73 20 6D 65 74 73 79 73 20 43 020F
73 75 20 74 73 65 74 20 65 63 69 76 021B
6F 63 65 72 20 61 74 61 64 20 72 65 0227
64 72 0233
31 0203
00000052 0235 308 MSG_SIZE=-ACC_MSG
0006 0235 309 ACC_MSG1: ; function code
02 0237 310 .WORD ACC$K_DISASEL
11 0238 311 .BYTE ACC$K_BATTRM ; batch job type
03 0239 312 .BYTE ACC$K_INSMESG ; arbitrary message type
04 023A 313 .BYTE ACC$K_INTTRM ; interactive job type
01 023B 314 .BYTE ACC$K_LOGTRM ; login failure termination type
10 023C 315 .BYTE ACC$K_PRCTRIM ; non-interactive process type
00 023D 316 .BYTE ACC$K_PRTJOB ; print job type
00000009 023E 317 .BYTE 0 ; terminator byte
00000052 023E 318 MSG1_SIZE=-ACC_MSG1
000001E3 0242 319 ACC_DESC: ; descriptor for accounting message
00000052 023E 320 .LONG MSG_SIZE
000001E3 0242 321 .ADDRESS ACC_MSG

```

	0246	323	OPTYPE:	
01000001	0246	324	.LONG	OPCSM_NM_CENTRL!<1024> ; opr type & ID table
02000002	024A	325	.LONG	OPCSM_NM_PRINT!<2024>
03000004	024E	326	.LONG	OPCSM_NM_TAPES!<3024>
04000008	0252	327	.LONG	OPCSM_NM_DISKS!<4024>
05000010	0256	328	.LONG	OPCSM_NM_DEVICE!<5024>
06001000	025A	329	.LONG	OPCSM_NM_OPER1!<6024>
07002000	025E	330	.LONG	OPCSM_NM_OPER2!<7024>
08004000	0262	331	.LONG	OPCSM_NM_OPER3!<8024>
09008000	0266	332	.LONG	OPCSM_NM_OPER4!<9024>
0A010000	026A	333	.LONG	OPCSM_NM_OPER5!<10024>
0B020000	026E	334	.LONG	OPCSM_NM_OPER6!<11024>
0C040000	0272	335	.LONG	OPCSM_NM_OPER7!<12024>
0D080000	0276	336	.LONG	OPCSM_NM_OPER8!<13024>
0E100000	027A	337	.LONG	OPCSM_NM_OPER9!<14024>
0F200000	027E	338	.LONG	OPCSM_NM_OPER10!<15024>
10400000	0282	339	.LONG	OPCSM_NM_OPER11!<16024>
11800000	0286	340	.LONG	OPCSM_NM_OPER12!<17024>
12000001	028A	341	.LONG	OPCSM_NM_CENTRL!<18024> ; just to make an even number
	028E	342	OPMSG_DESC:	
00000080	028E	343	.LONG	MSG_LEN ; SNDOPR msg buffer desc
00000296	0292	344	.ADDRESS	OPMSG
	0296	345	OPMSG:	
03	0296	346	.BYTE	OPCS_RQ_ROST ; function code
0000029A	0297	347	.BLKB	3 ; operator type
00000000	029A	348	.LONG	0 ; ID
00000316	029E	349	.BLKB	120 ; message or terminal info
00000080	0316	350	MSG_LEN=.	-OPMSG
	0316	351	SMSG_DESC:	
0000006C	0316	352	.LONG	SMSG_LEN ; SNDMSG msg buffer desc
0000031E	031A	353	.ADDRESS	SMSG
	031E	354	SMSG:	
0000	031E	355	.WORD	SMRSK_INITIAL ; SNDMSG msg buffer
00000330	0320	356	.BLKB	16 ; queue name
	0330	357	SMSG1:	
00000340	0330	358	.BLKB	16 ; device name
00000346	0340	359	.BLKB	6 ; file ID
0000034C	0346	360	.BLKB	6 ; directory ID
00000360	034C	361	.BLKB	20 ; filename
00000362	0360	362	.BLKB	2 ; job ID
0000036A	0362	363	.BLKB	8 ; job name
0000038A	036A	364	.BLKB	32 ; room for options and option data
0000006C	038A	365	SMMSG_LEN=.	-SMSG
	038A	366	SYM:	
0000039E	038A	367	.BLKB	20
	039E	368	:	
	039E	369	.ALIGN LONG	
	03A0	370	NAMBLK:	
	03A0	371	BNAM	
	0400	372	FAB:	
	0400	373	\$FAB	FAC=PUT,-
	0400	374		FNA=FILE NAME+1,-
	0400	375		FNS=COM FIL SIZ,-
	0400	376		NAM=NAMBLK,-
	0400	377		RAT=CR,-
	0400	378		RFM=VAR
	0450	379	RAB:	

SATSSS05
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:46:10 VAX/VMS Macro V04-00
E 8
R/W PSECT 5-SEP-1984 04:29:47 [UETPSY.SRC]SATSSS05.MAR;1 Page 9
(1)

0450	380	\$RAB	FAB=FAB,-
0450	381		MBF=1,-
0450	382		RBF=COM_FILE,-
0450	383		RSZ=RECO_SIZE
0494	384	FAB1:	
0494	385	\$FAB	FAC=PUT,-
0494	386		FNA=FILE_NAME1,-
0494	387		FNS=COM_FIL_SIZE

00000000 389 .PSECT SATSSS05,RD,WRT,EXE,LONG
0000 390 .SBTTL SATSSS05
0000 391 ++
0000 392 FUNCTIONAL DESCRIPTION:
0000 393
0000 394 After performing some initial housekeeping, such as
0000 395 printing the module begin message and acquiring needed privileges,
0000 396 the system services are tested in each of their normal conditions.
0000 397 Detected failures are identified and an error message is printed
0000 398 on the terminal. Upon completion of the test a success or fail
0000 399 message is printed on the terminal.
0000 400
0000 401 CALLING SEQUENCE:
0000 402
0000 403 \$ RUN SATSSS05 ... (DCL COMMAND)
0000 404
0000 405 INPUT PARAMETERS:
0000 406
0000 407 none
0000 408
0000 409 IMPLICIT INPUTS:
0000 410
0000 411 none
0000 412
0000 413 OUTPUT PARAMETERS:
0000 414
0000 415 none
0000 416
0000 417 IMPLICIT OUTPUTS:
0000 418
0000 419 Messages to SYSS\$OUTPUT are the only output from SATSSS05.
0000 420 They are of the form:
0000 421
0000 422 XUETP-S-SATSMS, TEST MODULE SATSSS05 BEGUN ... (BEGIN MSG)
0000 423 XUETP-S-SATSMS, TEST MODULE SATSSS05 SUCCESSFUL ... (END MSG)
0000 424 XUETP-E-SATSMS, TEST MODULE SATSSS05 FAILED ... (END MSG)
0000 425 XUETP-I-TEXT, ... (VARIABLE INFORMATION ABOUT A TEST MODULE FAILURE)
0000 426
0000 427 COMPLETION CODES:
0000 428
0000 429 The SATSSS05 routine terminates with a \$EXIT to the
0000 430 operating system with a status code defined by UETPS_SATSMS.
0000 431
0000 432 SIDE EFFECTS:
0000 433
0000 434 none
0000 435
0000 436
0000 437
0000 438 TEST_START SATSSS05 : let the test begin

```

0004'CF 0000 0000 . ENTRY SATSSS05,0
00 00 DD 0002 CLR  W^CURRENT_TC
0000'CF DF 0008 PUSHL #0
00000000'GF 02 FB 000C PUSHAL W^TPID
00000000'GF 00 FB 0013 CALLS #2,G^SYSSWAKE
0009'CF 7F 001A PUSHAQ W^TEST MOD NAME_D
00000000'GF 01 FB 001E CALLS #1,G^SPSSSETPRN
004C'CF 001F'CF DE 0028 BSBW W^MOD MSG PRINT
0044'CF 03 00 01 FO 002F MOVAL W^TEST MOD SUCC,W^TMD ADDR
00 00 DD 0036 INSV #SUCCESS,#0,#3,W^MOD_MSG_CODE
0AFE'CF 01 FB 0038 PUSHL #0
003D CALLS #1,W^REG_SAVE

STP0:
003D 439 .SBTTL SNDACC TESTS
003D 440 :+
003D 441 :-
003D 442 :$SNDACC tests
003D 443 : test ACC$K_NEWFILE
003D 444 : This function will not be tested because of the possible interference
003D 445 : that it might cause with the ACCOUNTNG.DAT file on a customer's system.
003D 446 : test ACC$K_INSMESG
003D 447 :-
003D 448 :-
003D 449 :-
003D 450 :-
003D 451 :-
59 00000000'9F DO 003D 452 MODE TO,10$,KRNL,NOREGS ; kernal mode to access PHD
0051'CF 69 DE 005A 453 MOVL #<TLSGL PHD,R9 ; get process header address
0171'CF 0031'CF DE 0061 454 MOVAL PHD$Q_PRIVMSK(R9),W^PRIVMASK ; get priv mask address
0177'CF 0134'CF DE 0066 455 MODE FROM,T0$ ; get back to user mode
00 00 DD 0067 456 PRIV ADD,OPER ; add the OPER priv.
0AFE'CF 01 FB 0087 457 MOVAL W^SNDACC,W^SERV_NAME ; set service name
0095 458 MOVAL W^UM,W^MODE ; set the mode
0097 459 PUSHL #0 ; push a dummy param
009C 460 CALLS #1,W^REG_SAVE ; save a reg snapshot
00AF 461 SCREMBX_S CHAN=W^MBCHAN ; create a mailbox
0082 462 BLBS R0,20$ ; br if OK
00BB 463 $EXIT_S R0 ; exit and show why
00BB 464 20$: ;-
00BB 465 $SNDACC_S MSGBUF=W^ACC_DESC,- ; try a ACC$K_NEWFILE
00BB 466 466 CHAN =W^MBCHAN ; check for success
00C8 467 FAIL_CHECK SSS_NORMAL ;-
00DB 468 PUSHL #SSS_NORMAL ; check the mailbox
00DB 469 CALLS #1,W^REG_CHECK ;-
00DB 470 :+
00DB 471 : test ACC$K_DISAACC
00DB 472 :-
00DB 473 :-
00DB 474 NEXT_TEST
00DB :+
0004'CF 01 DO 00DB STP1:
00 00 DD 00E0 MOVL #1,W^CURRENT_TC
PUSHL #0

```

0004'CF	01	FB	00E2		CALLS #1,W^REG_SAVE	
01E3'CF	04	BO	00E7	475	MOVW #ACCSK_DISAACC,W^ACC_MSG ; set the new function	
00A5'CF	0175'CF	SC	00EC	476	MOVZWL W^MBCHAN,W^SNDACC_CHAN ; set up the channel number	
00000000'8F			00F3	477	\$SNDACC_G W^SNDA ; try ACCSK_DISAACC with a little _G	
0B08'CF	01	FB	00FC	478	FAIL_CHECK SSS_NORMAL ; check for success	
OC0E'CF	00	FB	0102	479	PUSHL #SSS_NORMAL	
			0107	480	CALLS #1,W^REG_CHECK	
			010C	481	CALLS #0,W^READ_CHECK ; check the mailbox	
			010C	482	; test ACCSK_ENABACC	
			010C	483		
			010C	484		
			010C	485	; NEXT_TEST	
			010C			
0004'CF	02	DO	010C		STP2:	
00		DD	0111		MOVL #2,W^CURRENT_TC	
0AFE'CF	01	FB	0113		PUSHL #0	
01E3'CF	03	BO	0118	486	CALLS #1,W^REG_SAVE	
00000000'8F			011D	487	MOVW #ACCSK_ENABACC,W^ACC_MSG ; set function code	
0B08'CF	01	FB	0120	488	\$SNDACC_S MSGBOF=W^ACC_DESC,-	
OC0E'CF	00	FB	012D	489	CHAN =W^MBCHAN ; try ACCSK_ENABACC with a little _S	
			0130	490	FAIL_CHECK SSS_NORMAL ; check for success	
			0130	491	PUSHL #SSS_NORMAL	
			0130	492	CALLS #1,W^REG_CHECK	
			0130	493	CALLS #0,W^READ_CHECK ; check the mailbox	
			0130	494	; test ACCSK_DISASEL with all types selected	
			0130	495		
			0130	496		
			0130		; NEXT_TEST	
			0130			
0004'CF	03	DO	013D		STP3:	
00		DD	0142		MOVL #3,W^CURRENT_TC	
0AFE'CF	01	FB	0144		PUSHL #0	
0242'CF	0235'CF	DE	0149	497	CALLS #1,W^REG_SAVE	
023E'CF	09	DO	0150	498	MOVAL W^ACC_MSG1,W^ACC_DESC+4 ; set new message address	
00000000'8F			0155	499	MOVL #MSG1_SIZE,W^ACC_DESC ; set new message size	
0B08'CF	01	FB	015E	500	\$SNDACC_G W^SNDA ; try ACCSK_DISASEL	
OC0E'CF	00	FB	0164		FAIL_CHECK SSS_NORMAL ; check for success	
			0169	501	PUSHL #SSS_NORMAL	
			016E	502	CALLS #1,W^REG_CHECK	
			016E	503	CALLS #0,W^READ_CHECK ; check the mailbox	
			016E	504	; test ACCSK_ENABSEL	
			016E	505		
			016E	506		
			016E	507	; NEXT_TEST	
			016E			
0004'CF	04	DO	016E		STP4:	
00		DD	0173		MOVL #4,W^CURRENT_TC	
0AFE'CF	01	FB	0175		PUSHL #0	
0235'CF	05	BO	017A	508	CALLS #1,W^REG_SAVE	
00000000'8F			017F	509	MOVW #ACCSK_ENABSEL,W^ACC_MSG1 ; set new function	

SATSSS05
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:46:10 VAX/VMS Macro V04-00
SNDACC TESTS 5-SEP-1984 04:29:47 [UETPSY.SRC]SATSSS05.MAR;1

Page 13
(1)

00000000'8F	017F	510	CHAN =W^MBCHAN	: try ACC\$K_ENABSEL
OB08'CF	018F	511	FAIL_CHECK SSS_NORMAL	; check for success
OC0E'CF	01	FB	PUSHL #SSS NORMAL	
00	FB	0195	CALLS #1 W^REG_CHECK	
		512	CALLS #0,W^READ_CHECK	; check the mailbox

SAT
V04

```

019F 514 .SBTTL SNDERR_S TESTS
019F 515 :+
019F 516 :-
019F 517 : $SNDERR_S tests
019F 518 :-
019F 519 :-
019F 520 : NEXT_TEST
019F : STP5:
0004'CF 05 DD 019F MOVL #5,W^CURRENT_TC
0000'00 00 DD 01A4 PUSHL #0
0AFE'CF 01 FB 01A6 CALLS #1,W^REG_SAVE
0171'CF 0038'CF DE 01AB PRIV ADD,BUGCHK
01CB 521 MOVAL W^SNDERR,W^SERV_NAME ; add the BUGCHK priv.
01D2 522 SCREMBX_S CHAN=W^MBCHAN,- ; set service name
01D2 523 LOGNAME=W^MBNAM,-
01D2 524 PRMFLG=#0 ; make a mailbox
01E9 525 $GETCHN_S CHAN=W^MBCHAN,-
01E9 526 PRIBUF=W^MSGL ; get the unit number
00000000'GF 01 3C 01FF 528 MOVZWL W^BUF+DIB$W UNIT,-(SP)
FB 0204 529 CALLS #1,G^SYS$DERLMB ; push the MBX unit #
020B 530 $SNDERR_S MSGBUF=W^TEST_ERROR ; declare errorlog MBX
0216 531 FAIL_CHECK SSS_NORMAL ; try S form
00000000'BF 0216 PUSHL #SSS_NORMAL ; check for success
0808'CF 01 FB 021C CALLS #1,W^REG_CHECK

0071'CF 0175'CF 80 0221 532 GET1:
0221 533 MOVW W^MBCHAN,W^QIO+QIOS_CHAN ; get the channel number
0228 534 SQIO G W^QIO ; do a read
0231 535 SWAITFR_S EFN=#2 ; wait for it to complete
0075'CF 00'BF 88 023A 536 BISB2 #10SM NOW,W^QIO+QIOS_FUNC ; set the NOW modifier
00EB'CF 27 B1 0240 537 CMPW #EMBS$_SS,W^BUF+EMBS$ HD_ENTRY ; is this the right entry?
DA 12 0245 538 BNEQ GET1 ; br if not
56 00F9'CF DE 0247 539 MOVAL W^BUF+18,R6 ; set buffer address
57 01D5'CF DE 024C 540 MOVAL W^TEST_ERROR+8,R7 ; set good data address
58 00000064 8F DO 0251 541 MOVL #BUF_SIZE,R8 ; set byte count
0E1A'CF 00 FB 0258 542 CALLS #0,W^BUF_CHECK ; check results
025D 543 :+
025D 544 : $SNDERR_G tests
025D 545 :-
025D 546 :-
025D 547 :-
025D 548 : NEXT_TEST
025D : STP6:
0004'CF 06 DD 025D MOVL #6,W^CURRENT_TC
0000'00 DD 0262 PUSHL #0
0AFE'CF 01 FB 0264 CALLS #1,W^REG_SAVE
00E7'CF 00 2C 0269 549 MOVC5 #0,W^BUF,#0,#BUF_SIZE,W^BUF ; zero the buffer
00E7'CF 00 0272 550 PUSHL #0
0AFE'CF 01 DD 0275 551 CALLS #1,W^REG_SAVE ; push a dummy parameter
00E7'CF 00 FB 0277 552 $SNDERR_G W^SNDE ; save a reg snapshot
027C 553 FAIL_CHECK SSS_NORMAL ; try G
0285 554 PUSHL #SSS_NORMAL ; check for success
00000000'BF DD 0285 CALLS #1,W^REG_CHECK
0808'CF 01 FB 0288 555 GET2: BICL2 #10SM_NOW,W^QIO+QIOS_FUNC ; set to wait for mailbox
00000000'BF CA 0290
0299

```

SATSSS05
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:46:10 VAX/VMS Macro V04-00 Page 15
SNDRR_S TESTS 5-SEP-1984 04:29:47 [UETPSY.SRC]SATSSS05.MAR;1 (1)

0004'CF 07 DO 02C6 .SBTTL SNDOPR TESTS
 0004'CF 08 DO 0318
 0004'CF 09 DO 0366

0004'CF 00 DD 02C6
 0004'CF 01 FB 02CD
 0171'CF 003F'CF DE 02D2
 0296'CF 01 90 02D9
 0297'CF D4 02DE
 029A'CF FFFFFFFF 8F DO 02E2
 029E'CF B4 02EB
 02A0'CF 0239'CF 90 02EF
 02A1'CF 023A'CF DO 02F6
 00000000'8F DD 030D
 0B08'CF 01 FB 0313
 00000000'8F DD 035B
 0B08'CF 01 FB 0361
 0004'CF 00 DD 0310
 0004'CF 01 FB 031F
 0297'CF 00FFF01F 8F DO 0324
 029A'CF FFFFFFFF 8F DO 032D
 029E'CF 0154'CF B0 0336
 02A0'CF 014F'CF 90 033D
 02A1'CF 0150'CF DO 0344
 00000000'8F DD 0348
 0B08'CF 01 FB 034B
 00000000'8F DD 0358
 0B08'CF 01 FB 0360
 0004'CF 00 DD 0318
 0004'CF 01 FB 0318
 0004'CF 02 FB 0318
 0004'CF 03 FB 0318
 0004'CF 04 FB 0318
 0004'CF 05 FB 0318
 0004'CF 06 FB 0318
 0004'CF 07 DO 02C6
 0004'CF 08 DO 0318
 0004'CF 09 DO 0366

565 :+
 566 :+
 567 :+
 568 :+
 569 :+ \$SNDOPR tests
 570 :+
 571 :+ DISABLE tests with _S
 572 :+
 573 :+
 574 :+ NEXT_TEST

STP7:
 MOVL #7,W^CURRENT_TC
 PUSHL #0
 CALLS #1,W^REG_SAVE
 MOVAL W^SNDOPR,W^SERV NAME ; set service name
 MOVB #OPCS RQ TERME,0^OPMSG ; set the function code
 CLRL W^OPMSG+OPCSB MS ENAB ; set disable ID mask
 MOVL #-1,W^OPMSG+OPCSE MS MASK ; set operators to be disabled
 CLRW W^OPMSG+OPCSW MS DUNIT ; set unit to zero
 MOVB W^OPNAME,W^OPMSG+OPCST MS ONAME ; set operator name size
 MOVL W^OPNAME+1,W^OPMSG+OPCST_MS_ONAME+1 ; set operator device name
 SSNDOPR_S MSGBUF=W^OPMSG DESC,-
 CHAN=W^MBCHAN ; try S
 FAIL_CHECK SSS_NORMAL ; check success

575 :+
 576 :+
 577 :+
 578 :+
 579 :+
 580 :+
 581 :+
 582 :+
 583 :+
 584 :+
 585 :+
 586 :+
 587 :+ ENABLE tests with _S
 588 :+
 589 :+
 590 :+ NEXT_TEST

STP8:
 MOVL #8,W^CURRENT_TC
 PUSHL #0
 CALLS #1,W^REG_SAVE
 MOVL #ALL_OPR,W^OPMSG+OPCSB MS_ENAB ; set oprators to enable
 MOVL #-1,0^OPMSG+OPCSL MS_MASK ; set enableable bits
 MOVW W^TTUNIT,W^OPMSG+OPCSW MS_DUNIT ; set the terminal unit number
 MOVB W^TTNAM,W^OPMSG+OPCST MS_ONAME ; set the terminal name size
 MOVL W^TTNAM+1,W^OPMSG+OPCST_MS_ONAME+1 ; set the terminal name
 SSNDOPR_S MSGBUF=W^OPMSG DESC,-
 CHAN=W^MBCHAN ; enable the alternate terminal
 FAIL_CHECK SSS_NORMAL ; check for success

591 :+
 592 :+
 593 :+
 594 :+
 595 :+
 596 :+
 597 :+
 598 :+
 599 :+
 600 :+
 601 :+ REQUEST tests to make a request with ID = 1-18
 602 :+
 603 :+
 604 :+ NEXT_TEST

STP9:
 MOVL #9,W^CURRENT_TC

n 8

```

00B9'CF 00 DD 036B      PUSHL #0
          01 FB 036D      CALLS #1,W^REG_SAVE
0175'CF 01 3C 0372      MOVZWL W^MBCHAN,W^SNDO+SNDOPRS_CHAN ; set the channel number
0296'CF 03 90 0379      MOVB #OPCS_RQ_RQST,W^OPMSG ; set function code
024E'CF 2E 28 037E      MOVC3 #OP_MESG_LEN,W^OP_MESG,-
          57 D4 0386 605     W^OPMSG+OPCSL_MS_TEXT ; put the text in the message
          029A'CF D4 0388 606     CLRL R7 ; init loop variable
52 0246'CF DE 038C 607     CLRL W^OPMSG+OPCSL_MS_RQSTID ; init the ID field
          0297'CF 82 DO 0391 610     MOVAL W^OPTYPE,R2 ; set opr type list pointer
          00 DD 0396 611     10$:
          01 FB 0398 612     MOVL (R2)+,W^OPMSG+1 ; set opr type & ID
          039D 613     PUSHL #0 ; push a dummy parameter
          039D 614     CALLS #1,W^REG_SAVE ; save the registers
          03AD 615     $SNDOPR_S MSGBUF=W^OPMSG_DESC,-
          03AD 616     CHAN=W^MBCHAN ; try S form
          00000000'8F DD 03AD 617     FAIL_CHECK SSS_NORMAL ; check for success
          0B08'CF 01 FB 03B3 618     PUSHL #SSS_NORMAL
          0297'CF 82 DO 03B8 619     CALLS #1,W^REG_CHECK
          00 DD 03BD 620     MOVL (R2)+,W^OPMSG+1 ; set opr type & ID
          0AFE'CF 01 FB 03BF 621     PUSHL #0 ; push a dummy param
          03C4 622     CALLS #1,W^REG_SAVE ; save a reg snapshot
          03CD 623     $SNDOPR_G W^SNDO ; try G
          00000000'8F DD 03CD 624     FAIL_CHECK SSS_NORMAL ; check for success
          0B08'CF 01 FB 03D3 625     PUSHL #SSS_NORMAL
          B5 57 09 F2 03D8 626     CALLS #1,W^REG_CHECK ; do all opr types
          03DC 627     AOBLS S #9,R7,10$ ;+
          03DC 628     ; CANCEL tests to cancel requests 1-18
          03DC 629     ;-
          03DC 630     ;+
          03DC 631     ;+
          03DC 632     ;+
          03DC 633     ;+
          03DC 634     ;+
          03DC 635 10$:     ;+
          0004'CF 0A DO 03DC     ;+
          00 DD 03E1     ;+
          0AFE'CF 01 FB 03E3     ;+
          0296'CF 05 90 03E8 631     MOVL #10,W^CURRENT_TC
          52 0246'CF DE 03ED 632     PUSHL #0
          00008084 8F DD 03F2 633     CALLS #1,W^REG_SAVE
          57 D4 03F9 634     MOVB #OPCS_RQ_CANCEL,W^OPMSG
          03FB 635     MOVAL W^OPTYPE,R2 ; set function code
          03FB 636     MOVL #<OPCS_RQSTCAN>XFFFF>,R6 ; set table pointer
          0297'CF 82 DO 0400 637     CLRL R7 ; set completion code
          00 DD 0402 638     MOVL (R2)+,W^OPMSG+1 ; set loop variable
          0AFE'CF 01 FB 0402 639     PUSHL #0 ; set opr type & ID
          0407 639     CALLS #1,W^REG_SAVE ; push a dummy parameter
          0407 640     $SNDOPR_S MSGBUF=W^OPMSG_DESC,-
          0417 641     CHAN=W^MBCHAN ; save a reg snapshot
          00000000'8F DD 0417 642     FAIL_CHECK SSS_NORMAL ; try S form
          0B08'CF 01 FB 041D 643     PUSHL #SSS_NORMAL ; check for success
          0E59'CF 00 FB 0422 644     CALLS #1,W^REG_CHECK
          0297'CF 82 DO 0427 645     MOVL (R2)+,W^OPMSG+1 ; set opr type & ID
          00 DD 042C 645     PUSHL #0 ; push a dummy parameter
          0AFE'CF 01 FB 0433 646     CALLS #1,W^REG_SAVE ; save a reg snapshot
          0433 647     $SNDOPR_G W^SNDO ; try G form
          00000000'8F DD 043C 647     FAIL_CHECK SSS_NORMAL ; check success
          043C 647     PUSHL #SSS_NORMAL
    
```

0B08'CF 01 FB 0442
 0E59'CF 00 FB 0447 648 CALLS #1,W^REG_CHECK
 AB 57 09 F2 044C 649 CALLS #0,W^SND_CHECK
 :+ AOBLS 650 #9,R7,103 : check the results
 : do all opr types
 0450 651 : REPLY tests to respond to requests
 0450 652 :
 0450 653 :
 0450 654 :
 0450 655 : NEXT_TEST
 0450 :
 STP11:
 0004'CF 0B DO 0450 MOVL #11,W^CURRENT_TC
 00 DD 0455 PUSHL #0
 56 0A9E'CF 01 FB 0457 CALLS #1,W^REG_SAVE
 000008029 8F DO 045C 656 MOVL #<OPCS RQSfCMPLTE8^XFFFF>,R6 ; set expected status return
 0296'CF 04 90 0463 657 MOVB #OPCS RQ_REPLY W^OPMSG ; set the function
 0298'CF 56 80 0468 658 MOVW R6,W^OPMSG+OPCSW_MS_STATUS ; set status reply return
 029A'CF D4 046D 659 CLRL W^OPMSG+OPCSL_MS_RPCYID ; set the message ID
 029E'CF 0154'CF 80 0471 660 MOVW W^TTUNIT,W^OPMSG+OPCSW_MS_DUNIT ; set the unit number
 02A0'CF 014F'CF 05 28 0478 661 MOVC3 #5,W^TTNAME,W^OPMSG+OPCSST_MS_DNAME ; set the device name
 024E'CF 2E 28 0480 662 MOVC3 #OP MESG LEN,W^OP MESG,-
 02B0'CF 00 DD 0488 663 W^OPMSG+OPCSL_MS_OTEXT ; set the message text
 0A9E'CF 01 FB 048A 664 PUSHL #0 ; push a dummy parameter
 0EB1'CF 00 FB 048F 665 CALLS #1,W^REG_SAVE ; save a reg snapshot
 0494 666 CALLS #0,W^GENREQ ; generate a pending request
 0494 667 \$SNDOPR_S MSGBUF=W^OPMSG DESC,-
 0494 668 CHAN=W^MBCHAN ; try S
 04A4 669 FAIL_CHECK SSS_NORMAL ; check success
 00000000'8F DD 04A4
 0B08'CF 01 FB 04AA
 0E59'CF 00 FB 04AF 670 CALLS #0,W^SND_CHECK
 56 801C 8F BO 04B4 671 MOVW #<OPCS RQSTABORT8^XFFFF>,R6 ; check results
 0298'CF 56 80 04B9 672 MOVW R6,W^OPMSG+OPCSW_MS_STATUS ; set expected status return
 00 DD 04BE 673 PUSHL #0 ; set reply status code
 0A9E'CF 01 FB 04C0 674 CALLS #1,W^REG_SAVE ; push a dummy parameter
 0EB1'CF 00 FB 04C5 675 CALLS #0,W^GENREQ ; save a reg snapshot
 04CA 676 \$SNDOPR_G W^SNDO ; generate a pending request
 04D3 677 FAIL_CHECK SSS_NORMAL ; try G
 00000000'8F DD 04D3
 0B08'CF 01 FB 04D9
 0E59'CF 00 FB 04DE 678 CALLS #0,W^SND_CHECK ; check success
 56 8021 8F BO 04E3 679 MOVW #<OPCS RQSTPEND8^XFFFF>,R6 ; set expected status return
 0298'CF 56 80 04E8 680 MOVW R6,W^OPMSG+OPCSW_MS_STATUS ; set the reply status code
 00 DD 04ED 681 PUSHL #0 ; push a dummy parameter
 0A9E'CF 01 FB 04EF 682 CALLS #1,W^REG_SAVE ; save a reg snapshot
 0EB1'CF 00 FB 04F4 683 CALLS #0,W^GENREQ ; generate a pending request
 04F9 684 \$SNDOPR_S MSGBUF=W^OPMSG DESC,-
 04F9 685 CHAN=W^MBCHAN ; try S and leave the request pending
 0509 686 FAIL_CHECK SSS_NORMAL ; check success
 00000000'8F DD 0509
 0B08'CF 01 FB 050F
 0E59'CF 00 FB 0514 687 CALLS #0,W^SND_CHECK
 56 8084 8F BO 0519 688 MOVW #<OPCS RQSTCANC8^XFFFF>,R6 ; check results
 0298'CF 56 80 051E 689 MOVW R6,W^OPMSG+OPCSW_MS_STATUS ; set expected status return
 00 DD 0523 690 PUSHL #0 ; set reply status
 0A9E'CF 01 FB 0525 691 CALLS #1,W^REG_SAVE ; push a dummy parameter
 052A 692 \$SNDOPR_G W^SNDO ; save a reg snapshot
 : try _G

05B9 720 .SBTTL SND SMB TESTS
 05B9 721 :+
 05B9 722 :
 05B9 723 : \$SND SMB tests
 05B9 724 :
 05B9 725 : The following request types cannot be tested because of the lack of a
 05B9 726 : queueable device in the minimum configuration.
 05B9 727 :
 05B9 728 : SMRSK_ABORT, SMRSK_ASSIGN, SMRSK_JUSTIFY, SMRSK_ENTER
 05B9 729 :
 05B9 730 : test SMRSK_INITIAL by creating que1 and que2
 05B9 731 :
 05B9 732 :-
 05B9 733 : NEXT_TEST
 05B9 STP14:
 0004'CF 0E DD 05B9 MOVL #14,W^CURRENT_TC
 0000 00 DD 05BE PUSHL #0
 0AFA'CF 01 FB 05C0 CALLS #1,W^REG_SAVE
 0171'CF 0046'CF DE 05C5 734 MOVAL W^SND SMB, W^SERV NAME
 00C5'CF 0175'CF B0 05CC 735 MOVW W^MBCHAN, W^SND\$SND SMBS_CHAN : set service name
 53 0320'CF DE 05D3 736 MOVAL W^MSG+2,R3 : set the mailbox channel #
 63 01A1'CF 0E 28 05D8 737 MOVC3 #QUENAM1L W^QUENAM1,(R3) : set argument pointer
 53 0330'CF DE 05DE 738 MOVAL W^MSG1,R3 : set the queue name
 83 43 8F 90 05E3 739 MOVB #SMOSK_DETJOB,(R3)+ : set to proper end of que name
 63 94 05E7 740 CLRB (R3) : set to BATCH
 00 00 DD 05E9 741 PUSHL #0 : set option terminator
 0AFA'CF 01 FB 05EB 742 CALLS #1,W^REG_SAVE : push a dummy parameter
 05F0 743 \$SND SMB_S MSGBUF = W^MSG DESC,- : save a reg snapshot
 05F0 744 CHAN = W^MBCHAN
 0600 745 FAIL_CHECK SSS_NORMAL : try S INITIAL
 00000000'8F DD 0600 PUSHL #SSS NORMAL : check failure
 0B08'CF 01 FB 0606 CALLS #1,W^REG_CHECK
 56 00040001 8F DD 0608 746 MOVL #JBC\$ NORMAL,R6 : set expected return status
 0E59'CF 00 FB 0612 747 CALLS #0,W^SND CHECK : check results
 83 4E 8F 90 0617 748 MOVB #SMOSK_DisSwap,(R3)+ : set to disable swapping
 83 4D 8F 90 0618 749 MOVB #SMOSK_INIPRI,(R3)+ : set a new job priority
 83 02 90 061F 750 MOVB #2,(R3)+ : by default of 2
 83 4C 8F 90 0622 751 MOVB #SMOSK_JOBLIMIT,(R3)+ : set a job limit of
 83 02 90 0626 752 MOVB #2,(R3)+ : 2
 0320'CF 01AF'CF 0E 28 0628 754 CLRB (R3) : set the terminator
 00 00 DD 0633 755 MOVC3 #QUENAM2L W^QUENAM2, W^MSG+2 : set new que name
 0AFA'CF 01 FB 0635 756 PUSHL #0 : push a dummy parameter
 063A 757 CALLS #1,W^REG_SAVE : save a register snapshot
 0643 758 \$SND SMB_G W^SND\$: init the next que
 00000000'8F DD 0643 FAIL_CHECK SSS_NORMAL : check for failure
 0B08'CF 01 FB 0649 PUSHL #SSS NORMAL :
 0E59'CF 00 FB 064F CALLS #0,W^SND_CHECK : check the results
 0652 759 :+
 0652 760 :
 0652 761 : test SMRSK_START by starting que1 and que2
 0652 762 :
 0652 763 :
 0652 764 :-
 0652 765 : NEXT_TEST
 0652 STP15:

E 9

00000000'8F	DD 0719	PUSHL #SS\$_NORMAL	
OB08'CF 01	FB 071F	CALLS #1,W^REG_CHECK	
0E59'CF 00	FB 0724	CALLS #0,W^SND_CHECK	: check results
031E'CF 02	BO 0729	MOVW #SMRSK_START,W^SMSG	: release the request code
	072E	SSNDNSMB G W^SNDS	: reset the queue state
	0737	FAIL_CHECK SSS_NORMAL	: check failure
00000000'8F	DD 0737	PUSHL #SS\$_NORMAL	
OB08'CF 01	FB 073D	CALLS #1,W^REG_CHECK	
0E59'CF 00	FB 0742	CALLS #0,W^SND_CHECK	: check results
	0747	:+ test SMRSK_CREJOB, SMRSK_ADDFIL, SMRSK_CLSJOB	
	0747	0748 :-	
	0747	0749 :+ NEXT_TEST	
	0747	074A :+ STP18:	
0004'CF 12	DD 0747	MOVL #18,W^CURRENT_TC	
00	DD 074C	PUSHL #0	
DAFE'CF 01	FB 074E	CALLS #1,W^REG_SAVE	
	0753	SCREATE FAB = W^FAB	: open the command file
	075E	SCONNECT RAB = W^RAB	: connect up
	0769	SPUT RAB = W^RAB	: write the command file
0478'CF 02BC'CF	DE 0774	MOVAL W^REC1,W^RAB+RAB\$L RBF	: set rec #1 address
0472'CF 0B	BO 077B	MOVW #REC1 SIZE,W^RAB+RAB\$W_RSZ	: set rec #1 size
0478'CF 02C7'CF	DE 0788	SPUT RAB = W^RAB	: write record #1
0472'CF 29	BO 0792	MOVAL W^REC2,W^RAB+RAB\$L RBF	: set rec #2 address
	0797	MOVW #REC2 SIZE,W^RAB+RAB\$W_RSZ	: set rec #2 size
	07A2	SPUT RAB = W^RAB	: write record #2
	07AD	SDISCONNECT RAB = W^RAB	: disconnect
02F0'CF	DF 07B8	SCLOSE FAB = W^FAB	: file S05.COM now exists
02FD'CF	DF 07BC	PUSHAL W^OL1	: set option list #1
OCFD'CF 02	FB 07C0	PUSHAL W^JN1	: set job name #1
	07C5	CALLS #2,W^CRE_JOB	: put a job in the que
	07C5	:+ test SMRSK_ALTER on job #1 to release it	
	07C5	07C6 :-	
	07C5	07C6 :+ NEXT_TEST	
	07C5	07C6 :+ STP19:	
0004'CF 13	DO 07C5	MOVL #19,W^CURRENT_TC	
00	DD 07CA	PUSHL #0	
DAFE'CF 01	FB 07CC	CALLS #1,W^REG_SAVE	
031E'CF 0D	BO 07D1	MOVW #SMRSK_ALTER,W^SMSG	: set request code
53 0320'CF	DE 07D6	MOVAL W^MSGT2,R3	: set message buffer pointer
01A1'CF 0E	28 07DB	MOVC3 #QUENAM1L,W^QUENAM1,(R3)	: set the que name
53 0330'CF	DE 07E1	MOVAL W^MSG1,R3	: set to correct end of que name
83 OCE7'CF	BO 07E6	MOVW W^JOBID,(R3)+	: set job ID
83 22	90 0/EB	MOVB #SMOSK_JOBPRI,(R3)+	: set option code
83 01	90 07EE	MOVB #1,(R3)+	: set the job priority
63 94	07F1	CLRB (R3)	: terminate the option list
00	DD 07F3	PUSHL #0	: push a dummy parameter
DAFE'CF 01	FB 07F5	CALLS #1,W^REG_SAVE	: save a register snapshot
	07FA	SSNDNSMB G W^SNDS	try G ALTER
	0803	FAIL_CHECK SSS_NORMAL	: check failure

G 9

08CE 886 :-
08CE 887 : NEXT_TEST

0004'CF 16 DO 08CE
0AFE'CF 00 DD 08D3
0AFE'CF 01 FB 08D5
02F8'CF DF 08DA 888
0307'CF DF 08DE 889
0CFD'CF 02 FB 08E2 890
031E'CF 0C BO 08E7 891
0330'CF OCE7'CF BO 08EC 892
0332'CF 94 08F3 893
00000000'8F 00 DD 08F7 894
0AFE'CF 01 FB 08F9 895
08FE 896
0907 897

0004'CF 17 DO 0917
0AFE'CF 00 DD 091C
0AFE'CF 01 FB 091E
02F0'CF DF 0923 905
0311'CF DF 0927 906
0CFD'CF 02 FB 092B 907
53 031E'CF DE 0930 908
83 04 BO 0935 909
63 01AF'CF 0E 28 0938 910
53 0330'CF DE 093E 911
63 01A1'CF 0E 28 0943 912
02 A3 94 0949 913
00 DD 094C 914
0AFE'CF 01 FB 094E 915
0953 916
095C 917

0004'CF 17 DO 095C
0AFE'CF 00 FB 0962
0E59'CF 00 FB 0967 918
53 031E'CF DE 096C 919
83 0F BO 0971 920
63 01AF'CF 0E 28 0974 921
53 0330'CF DE 097A 922
83 OCE7'CF BO 097F 923
63 94 0984 924
00 DD 0986 925
0AFE'CF 01 FB 0988 926
098D 927
0996 928

STP22:
MOVL #2,W^CURRENT_TC
PUSHL #0
CALLS #1,W^REG_SAVE
PUSHAL W^OL2
PUSHAL W^JN2
CALLS #2,W^CRE_JOB
MOVW #SMRSK RAVJOB,W^SMSG
MOVW W^JOBID,W^SMSG1
CLRB W^SMSG1+2
PUSHL #0
CALLS #1,W^REG_SAVE
\$SND\$MB G W^SNDS
FAIL_CHECK SSS_NORMAL
PUSHL #SSS_NORMAL
CALLS #1,W^REG_CHECK
CALLS #0,W^SND_CHECK

: set option list #2
: set job name #2
: put job #2 in the que
: set request code
: set job ID
: set no options
: push a dummy parameter
: save a reg snapshot
: try G and nail the last job
: check failure

0917 899 :+
0917 900 : test SMRSK_MERGE on job #3
0917 901 :
0917 902 :
0917 903 :-
0917 904 :
0917 : NEXT_TEST

STP23:
MOVL #23,W^CURRENT_TC
PUSHL #0
CALLS #1,W^REG_SAVE
PUSHAL W^OL1
PUSHAL W^JN3
CALLS #2,W^CRE_JOB
MOVAL W^SMSG,R3
MOVW #SMRSK MERGE,(R3)+
MOVC3 #QUENAM2L,W^QUENAM2,(R3)
MOVAL W^SMSG1,R3
MOVC3 #QUENAM1L,W^QUENAM1,(R3)
CLRB 2(R3)
PUSHL #0
CALLS #1,W^REG_SAVE
\$SND\$MB G W^SNDS
FAIL_CHECK SSS_NORMAL
PUSHL #SSS_NORMAL
CALLS #1,W^REG_CHECK
CALLS #0,W^SND_CHECK
MOVAL W^SMSG,R3
MOVW #SMRSK RELEASE,(R3)+
MOVC3 #QUENAM2L,W^QUENAM2,(R3)
MOVAL W^SMSG1,R3
MOVW W^JOBID,(R3)+
CLRB (R3)
PUSHL #0
CALLS #1,W^REG_SAVE
\$SND\$MB G W^SNDS
FAIL_CHECK SSS_NORMAL

: set option list #3
: set job name #3
: put job 3 in the que
: set address
: set request code
: set queue name 1
: get to correct end of name
: set queue name 2
: set no options(*watch que name len
: push a dummy parameter
: save a reg snapshot
: try G MERGE
: check failure

0917 : check results

63 : check results
: set message address
: set request code
: set the que name
: get to the end of the quenam
: set the job ID
: set no options
: push a dummy parameter
: save a register snapshot
: release the job
: check for failures

SATSSS05
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) 16-SEP-1984 00:46:10 VAX/VMS Macro V04-00
SNDSMS TESTS 5-SEP-1984 04:29:47 [UETPSY.SRC]SATSSS05.MAR;1

Page 26
(2)

0AFE'CF	01	FB	0A97	971		CALLS #1,W^REG_SAVE	; save a reg snapshot
			0A9C	972		\$SNDSMB G W^SNDS	; delete the last que
			0AA5	973		FAIL_CHECK SSS_NORMAL	; check for failures
00000000'8F		DD	0AAB			PUSHL #SSS_NORMAL	
OB08'CF	01	FB	0AAB			CALLS #1,W^REG_CHECK	
OE59'CF	00	FB	0AB0	974		CALLS #0,W^SND_CHECK	; check the last results
			0ABS	975		SERASE FAB=W^FAB	; delete the .COM file
			0AC0	976		SERASE FAB=W^FAB1	; delete the .LOG file
			0ACB	977		\$DASSGN S CHAN=W^MBCHAN	
			0AD7	978		TEST_END	; drop the mailbox
004C'CF		DD	0AD7			PUSHL W^TMD_ADDR	
0048'CF		DD	0ADB			PUSHL W^TMN_ADDR	
02		DD	0ADF			PUSHL #2	
0044'CF		DD	0AE1			PUSHL W^MOD_MSG_CODE	
00000000'GF	04	FB	0AE5			CALLS #SST1_G^LIBSSIGNAL	
044'CF	01	1C	01	FO	0AEC	INSV #1,#SFSSV_INHIB_MSG,#1,W^MOD_MSG_CODE	
						PUSHL W^MOD_MSG_CODE	
0044'CF		DD	0AF3			CALLS #1,G^SYSSEXIT	
00000000'GF	01	FB	0AF7				

0008'CF 14 AD 28 28 0FFC
 50 04 AC D1 080A 1023 .SBTTL REG_SAVE
 0E 13 080E 1024 :++
 50 DD 0810 1025 : FUNCTIONAL DESCRIPTION:
 04 AC DD 0812 1026 : Subroutine to save R2-R11 in the register save location.
 0156'CF DF 0815 1027 .CALLING SEQUENCE:
 0B4A'CF 03 FB 0819 1028 PUSHL #0 ; save a dummy parameter
 081E 1029 081E 1029 10S: CALLS #1,W^REG_SAVE ; save R2-R11
 0008'CF 14 AD 28 29 081E 1030 .INPUT PARAMETERS:
 56 53 00000008'8F 22 13 0825 1031 NONE
 56 04 C3 0827 1032 .OUTPUT PARAMETERS:
 7E 56 02 C6 082F 1033 NONE
 51 53 CA 0832 1034 .--
 53 03 CA 0836 1035 REG_SAVE:
 0800 998 .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
 0807 999 MOVC3 #4+10,^X14(FP),W^REG_SAVE_AREA ; save the registers in the program
 0808 1000 RET
 0808 1001 .SBTTL REG_CHECK
 0808 1002 :++
 0808 1003 : FUNCTIONAL DESCRIPTION:
 0808 1004 : Subroutine to test R0 & R2-R11 for proper content after a service
 0808 1005 : execution. A snapshot is taken by the REG_SAVE routine at the
 0808 1006 : beginning of each step and this routine is executed after the
 0808 1007 : services have been executed.
 0808 1008 .CALLING SEQUENCE:
 0808 1009 PUSHL #SS\$_XXXXXX ; push expected R0 contents
 0808 1010 CALLS #1,W^REG_CHECK ; execute this routine
 0808 1011 .INPUT PARAMETERS:
 0808 1012 expected R0 contents on the stack
 0808 1013 .OUTPUT PARAMETERS:
 0808 1014 possible error messages printed using SPUTMSG
 0808 1015 .--
 0808 1016 REG_CHECK:
 0808 1017 WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
 0808 1018 CMPL 4(AP),R0 : is this the right fail code?
 0808 1019 BEQL 10\$: br if yes
 0808 1020 PUSHL R0 : push received data
 0808 1021 PUSHL 4(AP) : push expected data
 0808 1022 PUSHAL W^EXP : push the string variable
 0808 1023 CALLS #3,W^PRINT_FAIL : print the error message
 10S: CMPC3 #4+10,^X14(FP),W^REG_SAVE_AREA ; check all but R0
 BEQL 20\$; br if O.K.
 SUBL3 #REG_SAVE_AREA,R3,R6 ; calculate the register number
 DIVL2 #4,R8
 ADDB3 #^X2,R6,-(SP) ; set number past R0-R1 and save
 BICL2 #3,R1 ; backup to register boundrys
 BICL2 #3,R3

**F

```

61 DD 0B3C 1037 PUSHL (R1)
63 DD 0B3E 1038 PUSHL (R3)
00C9'CF DF 0B40 1039 PUSHAL W^REG
0B4A'CF 04 FB 0B44 1040 CALLS #4,W^PRINT_FAIL
0B49 1041 20$: : push received data
0B49 1042 RET : push expected data
0B4A 1043 .SBTTL PRINT_FAIL : set string pntr param.
0B4A 1044 :++ : print the error message
0B4A 1045 : FUNCTIONAL DESCRIPTION:
0B4A 1046 : Subroutine to report failures using $PUTMSG
0B4A 1047 :
0B4A 1048 : CALLING SEQUENCE:
0B4A 1049 : Mode #1 PUSHL EXPECTED Mode #2 PUSHL REG NUMBER
0B4A 1050 : PUSHL RECEIVED PUSHL EXPECTED
0B4A 1051 : PUSHAL STRING VAR PUSHL RECEIVED
0B4A 1052 : CALLS #3,W^PRINT_FAIL PUSHAL STRING VAR
0B4A 1053 : CALLS #4,W^PRINT_FAIL CALLS #4,W^PRINT_FAIL
0B4A 1054 : Mode #3 PUSHAL STRING VAR
0B4A 1055 : CALLS #1,W^PRINT_FAIL
0B4A 1056 :
0B4A 1057 : INPUT PARAMETERS:
0B4A 1058 : listed above
0B4A 1059 :
0B4A 1060 : OUTPUT PARAMETERS:
0B4A 1061 : an error message is printed using $PUTMSG
0B4A 1062 :
0B4A 1063 :--:
0B4A 1064 :
0B4A 1065 PRINT_FAIL:
0B4A 1066 WORD ^M<R2,R3,R4,R5>
0B4C 1067 SFAD_S W^CS1,W^MESSAGE,L,W^MSG,L,#TEST_MOD_NAME,W^SERV_NAME,W^CURRENT_TC
0B6D 1068 SPUTMSG_S W^MSGVEC ; print the message
04 6C 91 0B7E 1069 CMPB (AP),#4 ; is this a register message?
26 13 0B81 1070 BEQL 10S ; br if yes
01 6C 91 0B83 1071 CMPB (AP),#1 ; is this just a message?
48 13 0B86 1072 BEQL 20S ; br if yes
40 11 0B88 1073 SFAD_S W^CS2,W^MESSAGE,L,W^MSG,L,4(AP),8(AP),4(AP),12(AP) ; goto output message
0B8A? 1074 BRB 30S
0B8A? 1075 10$: SFAD_S W^CS3,W^MESSAGE,L,W^MSG,L,4(AP),16(AP),8(AP),4(AP),16(AP),12(AP) ; goto output message
19 11 0BCE 1076 BRB 30S
01D7'CF 04 AC 0D 0BD0 1078 20$: MOVL 4(AP),W^MSGVEC1+12 ; save string address
0BD6 1080 SPUTMSG_S W^MSGVEC1 ; print the message
11 11 0BE? 1081 BRB 40S ; skip the other message
0BE? 1082 30$: SPUTMSG_S W^MSGVEC ; print the message
0BE? 1083 40$: CALLS #0,W^MODE_ID ; identify the mode
0BFA 1084 40$: MOVAL W^TEST_MOD FAIL,W^THD_ADDR ; set failure message address
004C'CF 00 FB 0BFA 1085 INSV #ERROR,#0,73,W^MOD_MSG_CODE ; set severity code
002A'CF DE 0BFF 1086 RET
0044'CF 03 00 02 FO 0C06 1087 .SBTTL READ_CHECK
0COE 1088
0COE 1089
0COE 1090 :++ : FUNCTIONAL DESCRIPTION:
0COE 1091 : Subroutine to read a mailbox and check the status returned
0COE 1092 : from the SSNDACC system service.
0COE 1093 :

```

```

OCOE 1094 ; CALLING SEQUENCE:  

OCOE 1095 ;   CALLS #0,W^READ_CHECK  

OCOE 1096 ; INPUT PARAMETERS:  

OCOE 1097 ;   NONE  

OCOE 1098 ; OUTPUT PARAMETERS:  

OCOE 1099 ;   NONE  

OCOE 1100 ;  

OCOE 1101 ;  

OCOE 1102 ;  

OCOE 1103 ;  

OCOE 1104 ;--  

OCOE 1105 ;  

OCOE 1106 READ_CHECK:  

003C OCOE 1107 WORD  ^M<R2,R3,R4,R5>  

OC10 1108 $QIOW_S FUNC=#IOS READVBLK,-  

OC10 1109 CHAN=W^MBCHAN,-  

CC10 1110 IOSB=W^STATUSM,-  

OC10 1111 P1 =W^MBUF,-  

OC10 1112 P2 =#80 ; read the mail  

0000'8F 017B'CF B1 OC37 1113 CMPW W^MBUF,#MSG$_ACCRSP ; correct response type?  

13 13 OC3E 1114 BEQL 10$ ; br if yes  

017B'CF DD OC40 1115 PUSHL W^MBUF ; push received  

00000000'8F DD OC44 1116 PUSHL #MSG$_ACCRSP ; push expected  

0156'CF DF OC4A 1117 PUSHAL W^EXP ; push string variable  

FEF7 CF 03 FB OC4E 1118 CALLS #3,W^PRINT_FAIL ; print the failure  

OC53 1119 10$: CMPL W^MBUF+4,#4216!SS$_NORMAL ; check the results  

00040000'8F 017F'CF D1 OC53 1120 BEQL 20$ ; br if OK  

13 13 OC5C 1121 PUSHL W^MBUF+4 ; push received  

017F'CF DD OC5E 1122 PUSHL #4216!SS$_NORMAL ; push expected  

00040000'8F DD OC62 1123 PUSHAL W^EXP ; push the string variable  

0156'CF DF OC68 1124 CALLS #3,W^PRINT_FAIL ; print the failure  

FED9 CF 03 FB OC6C 1125 OC71 1126 20$: RET ; return  

04 OC71 1127

```

55 51 SF 54 41 42 SF 50 54 45 55 00
 31 45
 0D
 00000C8C
 00000CAC
 0000032
 0000003E
 00000CB4
 000A
 00000CC6
 00000CCC
 00000CD2
 07
 00000CE7
 0000
 00000CF1
 00
 0000003E

30 53 00
 00
 00000CE7
 0000
 00000CF1
 00
 0000003E

```

OC72 1129 .SBTTL CRE_JOB
OC72 1130 ++
OC72 1131 : FUNCTIONAL DESCRIPTION:
OC72 1132 : Routine to enter a job in queue #1
OC72 1133 :
OC72 1134 :
OC72 1135 : CALLING SEQUENCE:
OC72 1136 : PUSHAL W^OPTION_LIST : counted option list ending with a
OC72 1137 : byte of 0
OC72 1138 : PUSHAL W^JOB_NAME : counted job name ending with a byte of 0
OC72 1139 : CALLS #0,W^CRE_JOB : check buffer
OC72 1140 :
OC72 1141 : INPUT PARAMETERS:
OC72 1142 : Listed above plus initied NAMBLK to proper command file and
OC72 1143 : location MBCHAN initied to the mailbox channel.
OC72 1144 : OUTPUT PARAMETERS:
OC72 1145 : Location JOBID contains the job ID of the created job and
OC72 1146 : the job is placed in QUE #1
OC72 1147 :
OC72 1148 :--:
OC72 1149 :
OC72 1150 : CREATE: : create a job message buffer
00000032' OC72 1151 .LONG CR_MSGSIZ
00000C7A' OC76 1152 .ADDRESS .T4
0009 OC7A 1153 .WORD SMRSK_CREJOB
31 45 OC7C 1154 .ASCIC /UETP_BAT_QUE1/
00000C8C OC8A 1155 .BLKB 2
00000CAC OC8C 1156 OPTIONS: .BLKB 32
0000032 OCAC 1157 CR_MSGSIZ=-CREATE-8
0000003E' OCAC 1158 ADDFILE: : add a file message buffer
00000CB4' OCAC 1159 .LONG AD_MSGSIZ
000A OCB4 1160 .ADDRESS .T4
OCB4 1161 .WORD SMRSK_ADDFIL
00000CC6 OC86 1162 .BLKB 16
00000CCC OCC6 1163 DEVICE: .BLKB 6
00000CCC OCC6 1164 FID: .BLKB 6
00000CCC OCC6 1165 DID: .BLKB 6
00000CD2 OCCC 1166 .BLKB 6
00000CD2 OCD2 1167 .ASCIC /S05.COM/
07 OCD2 1168 .BLKB 13
00000CE7 OCDA 1169 .BLKB 13
0000 OCE7 1170 .WORD 0
00000CE7 OCE7 1171 JOBID: .BLKB 13
0000 OCE7 1172 .WORD 0
00000CF1 OCE9 1173 JOB_NAME: .BLKB 8
00 OCF1 1174 .BYTE 0
0000003E OCF2 1175 AD_MSGSIZ=-ADDFILE-8
00000003' OCF2 1176 .BLKB 8
00000CFA' OCF6 1177 CLOSE: : close a job message buffer
00000CFA' OCF6 1178 .LONG CL_MSGSIZ
000B OCF6 1179 .ADDRESS .T4
000B OCF6 1180 .WORD SMRSK_CLSJOB
00 OCF6 1181 .BYTE 0
00000003 OCFD 1182 CL_MSGSIZ=-CLOSE-8

```

		OCFD	1183	:	
		OCFD	1184	CRE_JOB:	
56	08	AC	07FC	WORD	"M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
FF80	CF	57	86	MOV _L	B(AP),R6
		57	86	MOVZBL	(R6)+R7
		66	57	MOV _{C3}	R7,(R6),W^OPTIONS
		56	04	MOVL	4(AP),R6
		57	86	MOVZBL	(R6)+R7
FFD0	CF	66	57	MOV _{C3}	R7,(R6),W^JOB_NAME
FFA5	CF	03C4	CF	MOV _{C3}	#FIDSIZ,W^NAMBLK+NAMSW_FID,W^FID
FFA3	CF	03CA	CF	MOV _{C3}	#DIDSIZ,W^NAMBLK+NAMSW_DID,W^DID
		56	03B4	MOVZBL	W^NAMBLK+NAMST_DVI,R6
FF7E	CF	66	57	INCL	R6
		03B4	CF	MOV _{C3}	R6,W^NAMBLK+NAMST_DVI,W^DEVICE
		56	00040001	MOVL	#JBCS_NORMAL,R6
		00	8F	PUSHL	#0
FDB8	CF	01	FB	CALLS	#1,W^REG_SAVE
				SSNDNSMB_S MSGBUF = W^CREATE,-	
				CHAN = W^MBCHAN	
00000000	'8F	DD	0D56	FAIL_CHECK	SSS_NORMAL
FDA7	CF	01	FB	PUSHL	#SSS_NORMAL
0E59	CF	00	FB	CALLS	#1,W^REG_CHECK
FF7A	CF	017D	CF	CALLS	#0,W^SND_CHECK
		BO	0D61	MOVW	W^MBUF+2,W^JOBID
			1203	SSNDNSMB_S MSGBUF = W^ADDFILE,-	
				CHAN = W^MBCHAN	
				FAIL_CHECK	SSS_NORMAL
				PUSHL	#SSS_NORMAL
00000000	'8F	DD	0D7D	CALLS	#1,W^REG_CHECK
FD80	CF	01	FB	CALLS	#0,W^SND_CHECK
0E59	CF	00	FB	SSNDNSMB_S MSGBUF = W^CLOSE,-	
			1208	CHAN = W^MBCHAN	
				FAIL_CHECK	SSS_NORMAL
				PUSHL	#SSS_NORMAL
00000000	'8F	DD	0D9D	CALLS	#1,W^REG_CHECK
FD60	CF	01	FB	CALLS	#0,W^SND_CHECK
0E59	CF	00	FB	RET	
		04	0DAD		
			1212		
			1213		

; get the option list pointer
; get the option list size
; set the option list
; get the job name pointer
; get the job name size
; set the job name
; set the FID
; set the DID
; get device name size
; include the count byte
; set the device name
; set expected status return
; set a dummy parameter
; save a reg snapshot
; create a job
; check for failure
; check the results
; save the job ID
; add the file
; check for failure
; check for failure
; check the results
; close the job
; check for failures
; check the results
; thats all folks

```

ODAE 1215 .SBTTL BUF_CHECK
ODAE 1216 :++ FUNCTIONAL DESCRIPTION:
ODAE 1217 : Routine to check the contents of a buffer against known good
ODAE 1218 : data.

ODAE 1219 : CALLING SEQUENCE:
ODAE 1220 : CALLS #0,W^BUF_CHECK ; check buffer
ODAE 1221 : INPUT PARAMETERS:
ODAE 1222 : R6 = buffer address
ODAE 1223 : R7 = good data address
ODAE 1224 : R8 = byte count
ODAE 1225 : OUTPUT PARAMETERS:
ODAE 1226 : NONE
ODAE 1227 : --
ODAE 1228 :
ODAE 1229 :
ODAE 1230 :
ODAE 1231 :
ODAE 1232 :-- BCSD:
ODAE 1233 : 00000050. ODAE 1234 BCSD:
ODAE 1235 : .LONG 80
ODAE 1236 : .ADDRESS BCBUF
ODAE 1237 : BCBUF: ODB2 1236 .ADDRESS BCBUF
ODAE 1238 : .BLKB 80
ODAE 1239 : BCOSD: ODB6 1237 .BLKB 80
ODAE 1240 : .LONG 0
ODAE 1241 : .ADDRESS BCBUF
ODAE 1242 : PARAM1: OEO6 1240 .LONG 0
ODAE 1243 : .BLKL 3
ODAE 1244 : OE1A 1241 .ADDRESS BCBUF
ODAE 1245 : BUF_CHECK: OE1A 1242 .BLKL 3
ODAE 1246 : .WORD ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10>
ODAE 1247 : MOVL R6,R9
ODAE 1248 : CMPC3 R8,(R7),(R6) ; save a copy of the buffer address
ODAE 1249 : BEQL 10$ ; check the buffer
ODAE 1250 : MOVAL B^PARAM1,R10 ; br if good
ODAE 1251 : MOVZBL (R3),(R10)+ ; set parameter pointer
ODAE 1252 : MOVZBL (R1),(R10)+ ; save bad data
ODAE 1253 : SUBL3 R9,R3,(R10)+ ; save good data
ODAE 1254 : MOVAL B^PARAM1,R10 ; save byte offset
ODAE 1255 : SFAD_S CTRSTR = W^CS6,- ; reset address pointer
ODAE 1256 : OUTLEN = W^BCOSD,-
ODAE 1257 : OUTBUF = W^BCSD,-
ODAE 1258 : P1 = (R10)+,-
ODAE 1259 : P2 = (R10)+,-
ODAE 1260 : P3 = (R10)+,-
FCF2 CF B3 AF DF OEA50 1261 PUSHAL B^BCOSD ; make the string
01 FB OEA53 1262 CALLS #1,W^PRINT_FAIL ; push the string variable
04 OEA58 1263 10$: RET ; print the failure
                                         ; return

```

OE59 1266 .SBTTL SND_CHECK
 OE59 1267 :++
 OE59 1268 : FUNCTIONAL DESCRIPTION:
 OE59 1269 : Routine to check the contents of a buffer against known good
 OE59 1270 : data.
 OE59 1271 : CALLING SEQUENCE:
 OE59 1272 : CALLS #0,W^SND_CHECK ; check buffer
 OE59 1273 : INPUT PARAMETERS:
 OE59 1274 : R6 = expected status code
 OE59 1275 : OUTPUT PARAMETERS:
 OE59 1276 : NONE
 OE59 1277 :
 OE59 1278 :
 OE59 1279 :
 OE59 1280 :
 OE59 1281 :--
 OE59 1282 :
 OE59 1283 SND_CHECK:
 003C OE59 1284 WORD ^M<R2,R3,R4,R5>
 OE5B 1285 \$QIOW_S FUNC=#IOS READVBLK,-
 OE5B 1286 CHAN=W^MBCHAN,-
 OE5B 1287 IOSB=W^STATUSM,-
 OE5B 1288 P1 =W^MBUF,-
 OE5B 1289 P2 =#80 : read the mail
 00000046'8F 0171'CF D1 OE82 1290 CMPL W^SERV_NAME,#SNDNSMB : SNDNSMB or SNDOPR
 0D 13 OE88 1291 BEQL 10\$: br if SNDNSMB
 56 017D'CF B1 OE8D 1292 CMPW W^MBUF+OPCSW_MS_STATUS,R6 : correct response type?
 1C 13 OE92 1293 BEQL 30\$: br if yes
 017D'CF DD OE94 1294 PUSHL W^MBUF+OPCSW_MS_STATUS : push received
 0B 11 OE98 1295 BRB 20\$: br to common code
 56 017F'CF D1 OE9A 1296 10\$: CMPL W^MBUF+4,R6 : correct status return?
 OF 13 OE9F 1297 BEQL 30\$: br if yes
 017F'CF DD OEA1 1298 PUSHL W^MBUF+4 : push received
 56 0156'CF DF OEA5 1300 20\$: PUSHL R6 : push expected
 FC9A CF 03 FB OEA7 1301 PUSHAL W^EXP : push string variable
 04 OEB0 1302 CALLS #3,W^PRINT_FAIL : print the failure
 OEB0 1303 30\$: RET
 OEB0 1304 4E: .SBTTL GENREQ
 OEB1 1305 SF: OEB1 1306 49: :++
 OEB1 1307 : FUNCTIONAL DESCRIPTION:
 OEB1 1308 : routine to generate a pending request for \$SNDOPR
 OEB1 1309 :
 OEB1 1310 : CALLING SEQUENCE:
 OEB1 1311 : CALLS #0,W^GENREQ ; generate a pending request
 OEB1 1312 :
 OEB1 1313 : INPUT PARAMETERS:
 OEB1 1314 : NONE
 OEB1 1315 :
 OEB1 1316 : OUTPUT PARAMETERS:
 OEB1 1317 : NONE
 OEB1 1318 :
 OEB1 1319 :
 OEB1 1320 :--
 OEB1 1321 :
 OEB1 1322 GENREQ:

SATSSS05
V04-000

- SATS SYSTEM SERVICE TESTS (SUCC S.C.) D 10
GENREQ 16-SEP-1984 00:46:10 VAX/VMS Macro V04-00
5-SEP-1984 04:29:47 [UETPSY.SRC]SATSSS05.MAR;1

Page 34
(2)

OFFC	0EB1	1323
	0EB2	1324
	0EB3	1325
	0EC3	1326
00000000'8F	DD	0EC3
FC3A CF 01	FB	0EC9
	04	0ECE 1327

```
WORD  ^M<R2,R3,R4,R5,R6,R7,R8,R9,R10,R11>
$SNDOPR_S MSGBUF = W^OP MSG1,-
CHAN = W^MBCHAN ; generate a request
FAIL_CHECK SSS_NORMAL ; check for failure
PUSHL #SSS_NORMAL
CALLS #1,W^REG_CHECK
RET
```

SAT
VO

41
41
41

65
61
6E

4B

53

OECF 1329 .SBTTL MODE_ID
OECF 1330 ++
OECF 1331 : FUNCTIONAL DESCRIPTION:
OECF 1332 : Subroutine to identify the mode that an exit handler is in.
OECF 1333 :
OECF 1334 : CALLING SEQUENCE:
OECF 1335 : CALLS #0,W^MODE_ID
OECF 1336 :
OECF 1337 : INPUT PARAMETERS:
OECF 1338 : MODE contains an address pointing to an ascii string desc.
OECF 1339 : of the current CPU mode.
OECF 1340 :
OECF 1341 : OUTPUT PARAMETERS:
OECF 1342 : NONE
OECF 1343 :
OECF 1344 :--
OECF 1345 :
OECF 1346 MODE_ID:
003C OECF 1347 WORD ^M<R2,R3,R4,R5>
OED1 1348 \$FAO_S W^CS5,W^MESSAGEL,W^MSGL,MODE ; format the error message
0EEA 1349 SPUTMSG_S W^MSGVEC ; print the mode message
04 OEFB 1350 RET

OEFC 1352 MOD_MSG_PRINT:
OEFC 1353 :
OEFC 1354 :
OEFC 1355 :
OEFC 1356 :
OEFC 1357 :
OEFC 1358 :
OEFC 1359 :
OEFC 1360 :
OEFC 1361 :
OF17 1362 :
OF18 1363 :
OF18 1364 :HMRTN:
OF18 1365 :
OF18 1366 :
OF18 1367 :
OF18 1368 :
OF18 1369 :
OF18 1370 :
OF18 1371 :
OF18 1372 :
OF18 1373 :
OF18 1374 :
OF18 1375 :
0000 00000059'FF 17 OF18 1376 :
OF1A 1377 :
OF20 1378 :
OF20 1379 :
OF20 1380 :
OF20 1381 :

* PRINTS THE TEST MODULE BEGUN/SUCCESSFUL/FAILED MESSAGES
* (USING THE PUTMSG MACRO).

PUTMSG <MOD_MSG_CODE,#2,TMN_ADDR,TMD_ADDR> : PRINT MSG
RSB : ... AND RETURN TO CALLER

* CHMRTN:
* CHANGE MODE ROUTINE. THIS ROUTINE GETS CONTROL WHENEVER
* A CMKRNL, CMEXEC, OR CMSUP SYSTEM SERVICE IS ISSUED
* BY THE MODE MACRO ('TO' OPTION). IT MERELY DOES
* A JUMP INDIRECT ON A FIELD SET UP BY MODE. IT HAS
* THE EFFECT OF RETURNING TO THE END OF THE MODE
* MACRO EXPANSION.

.WORD 0 : ENTRY MASK
JMP @CHM_CONT : RETURN TO MODE MACRO IN NEW MODE
* RET INSTR WILL BE ISSUED IN EXPANSION OF 'MODE FROM,' MACRO
.END SATSSS05

SS.TAB	= 00000494	R	03	EXP	= 00000156	R	02
SS.TABEND	= 000004E4	R	03	FAB	= 00000400	R	03
SS.TMP	= 00000000			FAB\$C_BID	= 00000003		
SS.TMP1	= 00000001			FAB\$C_BLN	= 00000050		
SS.TMP2	= 0000000CF			FAB\$C_SEQ	= 00000000		
SSARGS	= 00000002			FAB\$C_VAR	= 00000002		
SST1	= 00000004			FAB\$L_ALQ	= 00000010		
SST2	= 00000004			FAB\$L_FOP	= 00000004		
A				FAB\$V_CHAN_MODE	= 00000002		
ACCSK_BATTRM	= 00000064			FAB\$V_CR	= 00000001		
ACCSK_DISAACC	= 00000002			FAB\$V_FILE_MODE	= 00000004		
ACCSK_DISASEL	= 00000004			FAB\$V_LNM_MODE	= 00000000		
ACCSK_ENABACC	= 00000006			FAB\$V_PUT	= 00000000		
ACCSK_ENABSEL	= 00000003			FAB\$W_GBC	= 00000048		
ACCSK_INSMESG	= 00000005			FAB1	00000494	R	03
ACCSK_INSMESG	= 00000001			FID	00000CC6	R	04
ACCSK_INNSMSG	= 00000011			FIDSIZ	= 00000006		
ACCSK_INNSMSG	= 00000003			FILE_NAME	0000027C	R	02
ACCSK_INTTRM	= 00000004			FILE_NAME1	00000290	R	02
ACCSK_LOGTRM	= 00000001			FILNAMSIZ	= 00000014		
ACCSK_PRCTRIM	= 00000010			GENREQ	00000EB1	R	04
ACCSK_PRTJOB	= 0000023E	R	03	GET1	00000221	R	04
ACC_DESC	000001E3	R	03	GET2	00000299	R	04
ACC_MSG	00000235	R	03	INFO	= 00000003		
ACC_MSG1	00000CAC	R	04	IOSM_NOW	*****	X	04
ADDFILE	= 0000003E			IOS READVBLK	*****	X	03
AD MSGSIZ	= 00FF01F			JBCS_NORMAL	= 00040001		
ALC_OPR	= 00000164	R	02	JN1	000002FD	R	02
BAT_IMP_EXC	00000DB6	R	04	JN2	00000307	R	02
BCBDF	00000E06	R	04	JN3	00000311	R	02
BCOSD	00000DAE	R	04	JOBID	00000CE7	R	04
BCSD				JOB_NAME	00000CE9	R	04
BUF	000000E7	R	03	LF	= 0000000A		
BUF_CHECK	00000E1A	R	04	LIB\$SIGNAL	*****	X	04
BUF_SIZE	= 00000064			MBCHAN	00000175	R	03
CHMRTN	00000F18	R	04	MBNAM	00000140	R	02
CHM_CONT	00000059	R	03	MBUF	0000017B	R	03
CLOSE	00000CF2	R	04	MESSAGE	00000169	R	03
CL_MSGSIZ	= 00000003			MODE	00000177	R	03
COM_FILE	00000297	R	02	MODE_ID	00000ECF	R	04
COM_FIL_SIZ	= 00000007			MOD_MSG_CODE	00000044	R	03
CR	= 0000000D			MOD_MSG_PRINT	00000EFC	R	04
CREATE	00000C72	R	04	MSG\$ACCRSP	*****	X	04
CRE_JOB	00000CFD	R	04	MSG1	= 0000036		
CR_MSGSIZ	= 00000032			MSG1_SIZE	= 00000009		
CST	0000004D	R	02	MSGL	000000DF	R	03
CS2	0000007F	R	02	MSGVEC	000001BD	R	02
CS3	000000AC	R	02	MSGVEC1	000001CB	R	03
CS5	000000DF	R	02	MSG_LEN	= 00000080		
CS6	000000F4	R	02	MSG_SIZE	= 00000052		
CTL\$GL_PHD	*****	X	04	NAMSB_ESS	= 0000000A		
CURRENT_TC	00000004	R	03	NAMSB_NOP	= 00000008		
DEVICE	00000CB6	R	04	NAMSB_RSS	= 00000002		
DIBSW_UNIT	= 0000000C			NAMSC_BID	= 00000002		
DID	00000CCC	R	04	NAMSC_BLN	= 00000060		
DIDSIZ	= 00000006			NAMSL_ESA	= 0000000C		
EMBSC_SS	= 00000027			NAMSL_RSA	= 00000004		
EMBSW_HD_ENTRY	= 00000004						
ERROR	= 00000002						

NAMST_DVI	= 00000014		PRVSV_OPER	= 00000012	
NAMSW_DID	= 0000002A		PRVPRT	= 00000050	R 03
NAMSW_FID	= 00000024		QIO	= 00000069	R 03
NAMBLR	= 000003A0 R	03	QIOS_ASTADR	= 00000014	
NAME_SIZE	= 00000008		QIOS_ASTPRM	= 00000018	
OL1	= 000002F0 R	02	QIOS_CHAN	= 00000008	
OL1S	= 00000007		QIOS_EFN	= 00000004	
OL2	= 000002F8 R	02	QIOS_FUNC	= 0000000C	
OL2S	= 00000004		QIOS_IOSB	= 00000010	
OPCSB_MS_ENAB	= 00000001		QIOS_NARGS	= 0000000C	
OPCSL_MS_MASK	= 00000004		QIOS_P1	= 0000001C	
OPCSL_MS_OTEXT	= 0000001A		QIOS_P2	= 00000020	
OPCSL_MS_RPLYID	= 00000004		QIOS_P3	= 00000024	
OPCSL_MS_RQSTID	= 00000004		QIOS_P4	= 00000028	
OPCSL_MS_TEXT	= 00000008		QIOS_P5	= 0000002C	
OPCSM_NM_CENTRL	= 00000001		QIOS_P6	= 00000030	
OPCSM_NM_DEVICE	= 00000010		QUENAM1	= 000001A1 R	02
OPCSM_NM_DISKS	= 00000008		QUENAM1L	= 0000000E	
OPCSM_NM_OPER1	= 00001000		QUENAM2	= 000001AF R	02
OPCSM_NM_OPER10	= 00200000		QUENAM2L	= 0000000E	
OPCSM_NM_OPER11	= 00400000		RAB	= 00000450 R	03
OPCSM_NM_OPER12	= 00800000		RAB\$B_RAC	= 0000001E	
OPCSM_NM_OPER2	= 00002000		RAB\$C_BID	= 00000001	
OPCSM_NM_OPER3	= 00004000		RAB\$C_BLN	= 00000044	
OPCSM_NM_OPER4	= 00008000		RAB\$C_SEQ	= 00000000	
OPCSM_NM_OPER5	= 00010000		RABSL_CTX	= 00000018	
OPCSM_NM_OPER6	= 00020000		RABSL_RBF	= 00000028	
OPCSM_NM_OPER7	= 00040000		RABSL_ROP	= 00000004	
OPCSM_NM_OPER8	= 00080000		RABSW_RSZ	= 00000022	
OPCSM_NM_OPER9	= 00100000		READ_CHECK	= 00000C0E R	04
OPCSM_NM_PRINT	= 00000002		RECO_SIZE	= 00000025	
OPCSM_NM_TAPES	= 00000004		REC1	= 000002BC R	02
OPCST_MS_ONAME	= 0000000A		REC1_SIZE	= 0000000B	
OPCSW_MS_OUNIT	= 00000008		REC2	= 000002C7 R	02
OPCSW_MS_STATUS	= 00000002		REC2_SIZE	= 00000029	
OPCS_RQSTABORT	= 0005801C		REG	= 000000C9 R	03
OPCS_RQSTCAN	= 00058084		REGNUM	= 000000DB R	03
OPCS_RQSTCMPLTE	= 00058029		REG_CHECK	= 00000B08 R	04
OPCS_RQSTPEND	= 00058021		REG_SAVE	= 00000AFE R	04
OPCS_RQ_CANCEL	= 00000005		REG_SAVE_AREA	= 00000008 R	03
OPCS_RQ_REPLY	= 00000004		RETADR	= 0000005D R	03
OPCS_RQ_RQST	= 00000003		SATSSS05	= 00000000 RG	04
OPCS_RQ_TERME	= 00000001		SERV_NAME	= 00000171 R	03
OPMSG	00000296 R	03	SEVERE	= 00000004	
OPMSG_DESC	0000028E R	03	SHRSK SHRDEF	= 00000001	
OPNAME	00000239 R	02	SHRS TEXT	= 00001130	
OPTIONS	00000C8C R	04	SMOSK_DETJOB	= 00000043	
OPTYPE	00000246 R	03	SMOSK_DISWAP	= 0000004E	
OP_MESG	0000024E R	02	SMOSK_HOLD	= 00000021	
OP_MESG_LEN	= 0000002E		SMOSK_INIPRI	= 0000004D	
OP_MSG1	0000023E R	02	SMOSK_JOBBLIM	= 0000004C	
PARAM1	00000E0E R	04	SMOSK_JOBPRI	= 00000022	
PHDSQ_PRIVMSK	= 00000000		SMOSK_PARAMS	= 00000026	
PRINT_FAIL	00000B4A R	04	SMRSK_ADDFIL	= 0000000A	
PRIVMASK	00000051 R	03	SMRSK_ALTER	= 0000000D	
PRIV_ARGS	= 00000002		SMRSK_CLSJOB	= 0000000B	
PRVSV_BUGCHK	= 00000017		SMRSK_CREJOB	= 00000009	

SMRSK_DELETE	= 00000001		STP5	0000019F R	04
SMRSK_INITIAL	= 00000000		STP6	0000025D R	04
SMRSK_MERGE	= 00000004		STP7	000002C6 R	04
SMRSK_PAUSE	= 00000003		STP8	00000318 R	04
SMRSK_RELEASE	= 0000000F		STP9	00000366 R	04
SMRSK_RMVJOB	= 0000000C		STSSV_INHIB_MSG	= 000001C	
SMRSK_START	= 00000002		SUCCESS	= 00000001	
SMRSK_STOP	= 00000007		SYM	0000038A R	03
SMRSK_SYNCJOB	= 00000011		SYM_DESC	00000195 R	02
SMSG	0000031E R	03	SYM_NAME	00000192 R	02
SMSG1	00000330 R	03	SYSSCLOSE	***** GX	04
SMSG_DESC	00000316 R	03	SYSSCMKRL	***** GX	04
SMSG_LEN	= 0000006C		SYSSCONNECT	***** GX	04
SNDA	0000009D R	03	SYSSCREATE	***** GX	04
SNDACC	00000031 R	02	SYSSCREMBX	***** GX	04
SNDACCS_CHAN	= 00000008		SYSSDASSGN	***** GX	04
SNDACCS_MSGBUF	= 00000004		SYSSDELLOG	***** GX	04
SNDACCS_NARGS	= 00000002		SYSSDERLMB	***** X	04
SNDE	000000A9 R	03	SYSSDISCONNECT	***** GX	04
SNDERR	00000038 R	02	SYSSERASE	***** GX	04
SNDERRS_MSGBUF	= 00000004		SYSEXIT	***** GX	04
SNDERRS_NARGS	= 00000001		SYSSFAO	***** X	04
SNDO	000000B1 R	03	SYSSGETCHN	***** GX	04
SNDOPR	0000003F R	02	SYSSHIBER	***** GX	04
SNDOPRS_CHAN	= 00000008		SYSSPUT	***** GX	04
SNDOPRS_MSGBUF	= 00000004		SYSSPUTMSG	***** GX	04
SNDOPRS_NARGS	= 00000002		SYSSQIO	***** GX	04
SNDS	000000BD R	03	SYSSQIOW	***** GX	04
SNDSMB	00000046 R	02	SYSSSETPRN	***** GX	04
SNDSMBS_CHAN	= 00000008		SYSSSETPRV	***** GX	04
SNDSMBS_MSGBUF	= 00000004		SYSSNDACC	***** GX	04
SNDSMBS_NARGS	= 00000002		SYSSNDERR	***** GX	04
SND_CHECK	00000E59 R	04	SYSSNDOPR	***** GX	04
SSS_NORMAL	***** X	04	SYSSNDSMB	***** GX	04
STATUS	000001DB R	03	SYSTRNLOG	***** GX	04
STATUSM	00000065 R	03	SYSSWAITFR	***** GX	04
STEP	= 00000018		SYSSWAKE	***** GX	04
STPO	0000003D R	04	TEST_ERROR	000001CD R	02
STP1	000000DB R	04	TEST_MOD_BEGIN	00000019 R	02
STP10	000003DC R	04	TEST_MOD_FAIL	0000002A R	02
STP11	00000450 R	04	TEST_MOD_NAME	00000000 R	02
STP12	00000543 R	04	TEST_MOD_NAME_D	00000009 R	02
STP13	00000575 R	04	TEST_MOD_SUCC	0000001F R	02
STP14	000005B9 R	04	TMD_ADDR	0000004C R	02
STP15	00000653 R	04	TMN_ADDR	00000048 R	02
STP16	000006A9 R	04	TPID	00000000 R	03
STP17	000006F8 R	04	TTNAM	0000014F R	02
STP18	00000747 R	04	TTUNIT	00000154 R	02
STP19	000007C5 R	04	UETPS_SATSMS	= 007480D9	
STP2	0000010C R	04	UETPS_TEXT	= 00741133	
STP20	00000813 R	04	UM	00000134 R	02
STP21	00000848 R	04	WARNING	= 00000000	
STP22	000008CE R	04	YES	0000019D R	02
STP23	00000917 R	04	YES_DESC	0000018A R	02
STP24	00000A0A R	04			
STP3	0000013D R	04			
STP4	0000016E R	04			

+-----+
! Psect synopsis !
+-----+

PSECT name

	Allocation	PSECT No.	Attributes																
ABS .	00000000 (0.)	00 (0.)	NOPIC	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE						
\$ABSS	00000000 (0.)	01 (1.)	NOPIC	USR	CON	ABS	LCL	NOSHR	EXE	RD	WRT	NOVEC	BYTE						
RODATA	0000031B (795.)	02 (2.)	NOPIC	USR	CON	REL	LCL	NOSHR	NOEXE	RD	NOWRT	NOVEC	LONG						
RWDATA	000004E4 (1252.)	03 (3.)	NOPIC	USR	CON	REL	LCL	NOSHR	NOEXE	RD	WRT	NOVEC	LONG						
SATSSS05	00000F20 (3872.)	04 (4.)	NOPIC	USR	CON	REL	LCL	NOSHR	EXE	RD	WRT	NOVEC	LONG						

+-----+
! Performance indicators !
+-----+

Phase

Phase	Page faults	CPU Time	Elapsed Time
Initialization	37	00:00:00.08	00:00:00.43
Command processing	152	00:00:00.73	00:00:03.13
Pass 1	589	00:00:24.73	00:00:38.18
Symbol table sort	0	00:00:02.46	00:00:02.81
Pass 2	297	00:00:05.77	00:00:07.90
Symbol table output	39	00:00:00.27	00:00:00.28
Psect synopsis output	3	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	1119	00:00:34.06	00:00:52.76

The working set limit was 2000 pages.

144825 bytes (283 pages) of virtual memory were used to buffer the intermediate code.

There were 90 pages of symbol table space allocated to hold 1646 non-local and 19 local symbols.

1381 source lines were read in Pass 1, producing 40 object records in Pass 2.

98 pages of virtual memory were used to define 88 macros.

+-----+
! Macro library statistics !
+-----+

Macro library name

Macros defined

-\$255\$DUA28:[SHRLIB]UETP.MLB:1	12
-\$255\$DUA28:[SYS.OBJ]LIB.MLB:1	6
-\$255\$DUA28:[SYSLIB]STARLET.MLB:2	67
TOTALS (all libraries)	85

2203 GETS were required to define 85 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LIS\$:\$SATSSS05/OBJ=OBJ\$:\$SATSSS05 MSRC\$:\$SATSSS05/UPDATE=(ENH\$:\$SATSSS05)+EXECMLS/LIB+SHRLIB\$:\$UETP/LIB

0421 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

